Aplicación de Algoritmos de Machine Learning para la Predicción de Diabetes y Anemia

Proyecto 1 - Inteligencia Artificial - Steven Pacheco - IIS2024

Daniela Alvarado Andrade
Instituto Tecnológico de Costa Rica
Ingeniería en Computación - 2021004342
dani.alvarado@estudiantec.cr

Alexia Denisse Cerdas Aguilar
Instituto Tecnológico de Costa Rica
Ingeniería en Computación - 2019026961
acerdas1701@estudiantec.cr

***Abstract*—This study explores the application of machine learning algorithms for predicting two prevalent medical conditions: diabetes and anemia. Using logistic regression and K-nearest neighbors (KNN), we applied these models to two separate datasets for binary classification. For the diabetes dataset, logistic regression, enhanced by ElasticNet regularization, outperformed KNN in overall accuracy, precision, and recall, with an AUC-ROC of 0.8728. Meanwhile, KNN demonstrated high recall but was more prone to overfitting. In the anemia dataset, both models yielded strong performance, but logistic regression showed superior stability and reduced risk of overfitting when applied to the balanced dataset using synthetic minority oversampling (SMOTE). The findings support logistic regression as the more reliable model for clinical predictions, while KNN requires careful tuning to avoid overfitting.**

***Index Terms*—Health, diabetes, anemia, machine learning, logistic regression, k-nearest neighbors, binary classification, SMOTE**

I. Introducción

La inteligencia artificial se está convirtiendo en una parte integral en el campo de la salud con la ayuda de algoritmos que apoyan a los profesionales médicos en entornos clínicos y en investigaciones en curso. Los modelos de machine learning tienen el potencial de monitorear signos vitales de los pacientes, alertar en caso de aumento de ciertos factores de riesgo e incluso predecir complicaciones antes de que se manifiesten. Además, pueden mejorar la atención con el paciente, reducir errores, optimizar tratamientos y ayudar a la gestión de recursos hospitalarios [1]. Esta investigación se centrará en dos enfermedades de alta prevalencia: la diabetes y la anemia.

La diabetes es una enfermedad metabólica crónica caracterizada por niveles elevados de glucosa en sangre (o azúcar en sangre), que con el tiempo conduce a daños graves en el corazón, los vasos sanguíneos, los ojos, los riñones y los nervios [2]. Por otro lado, la anemia es una afección que se desarrolla cuando la sangre produce una cantidad inferior a la normal de glóbulos rojos sanos lo que genera cansancio o debilidad por la falta de oxígeno, problemas al respirar, mareos, latidos cardíacos irregulares y demás [3].

Para cada enfermedad se cuenta con un conjunto de datos que será evaluado para predecir si una persona padece o no alguna de las condiciones. El conjunto de datos sobre diabetes [4] fue asignado por el profesor, el cual cuenta con 768 datos. Mientras que el de anemia [5] cuenta con 1421 datos, fue seleccionado por las integrantes del proyecto, con el objetivo de mantener el enfoque de la investigación en el ámbito médico e indagar en la exploración de este tipo de datos.

El objetivo principal de este estudio consiste en aplicar algoritmos como Regresión Logística y K-Nearest Neighbors (KNN) en los dos conjuntos de datos de clasificación binaria, al igual que explorar, analizar y evaluar el rendimiento de estos modelos. Esto permitirá comparar la eficacia de los algoritmos y comprender su comportamiento en distintos escenarios, fortaleciendo así la comprensión teórica y práctica de los métodos de machine learning. Para esto, el documento estará dividido en metodología donde se explicará el cómo de los procesos realizados y respectivos análisis, resultados de ambos datasets, discusión sobre los resultados y por último las conclusiones finales.

A. Hipótesis Diabetes Dataset

Basándonos en las características del conjunto de datos y en el comportamiento de los modelos de clasificación, se plantea la siguiente hipótesis: se espera que el modelo de regresión logística ofrezca un mejor rendimiento general en términos de precisión y balance de clases en comparación con KNN. Esto se debe a que la regresión logística es particularmente eficiente en problemas binarios, como el diagnóstico de diabetes, donde las relaciones entre las variables predictoras y la variable de clase suelen ser lineales o casi lineales.

Por otro lado, se espera que el modelo KNN con un valor de k óptimo ofrezca un rendimiento competitivo, pero con mayor sensibilidad a la escala y distribución de los datos. Se anticipa que un valor de k en el rango de 5 a 7 podría proporcionar los mejores resultados, dado que valores más bajos podrían llevar a un sobreajuste mientras que valores más altos podrían diluir la capacidad del modelo para identificar correctamente los patrones en los datos.

B. Hipótesis Anemia Dataset

Se postula que la Regresión Logística es el modelo más adecuado para el conjunto de datos de clasificación binaria en comparación con K-Nearest Neighbors (KNN). Esta hipótesis se fundamenta en el hecho de que la Regresión Logística está diseñada específicamente para problemas de clasificación binaria [6], mientras que el KNN generalmente se usa como un algoritmo de clasificación, partiendo de que se toma el promedio de los k vecinos más cercanos para hacer una predicción sobre una clasificación [7].

Además, dado que el conjunto de datos contiene una cantidad significativa de ejemplos, se anticipa que la Regresión Logística, al ser un modelo que se ajusta bien a grandes cantidades de datos, tiene una mayor capacidad para generalizar y evitar el sobreajuste. Por lo tanto, se espera que la Regresión Logística, por su capacidad inherente para manejar problemas de clasificación binaria y su robustez en conjuntos de datos grandes, supere a KNN en términos de métricas de evaluación como el AUC-ROC, la precisión, el recall y el F1-score.

Sin embargo, se considera que la inclusión de características adicionales, como la edad de los pacientes, podría mejorar significativamente la capacidad predictiva del modelo, ya que variables como esta suelen tener un impacto importante en la aparición de condiciones médicas como la anemia.

II. Metodología

En esta sección, se describen los datasets utilizados para el análisis. Los datos son fundamentales para la investigación, ya que permiten aplicar y evaluar los algoritmos seleccionados. A continuación, se presentan detalles sobre cada uno de los conjuntos de datos empleados.

A. *Diabetes Dataset*

1) Carga del conjunto de datos: El primer paso consistió en la carga del conjunto de datos en un entorno de análisis utilizando la librería Pandas. El archivo CSV fue cargado y se realizó una inspección preliminar para verificar la estructura del conjunto de datos, observar los primeros registros y obtener una idea general de las variables disponibles. Esto permitió asegurar que el archivo de datos estuviera correctamente formateado y que el análisis posterior fuera coherente con la estructura de las variables.

El conjunto de datos incluye 768 registros, donde cada uno representa a una paciente, y un total de 9 columnas, que consisten en las características biométricas y la variable objetivo Outcome, la cual indica si una paciente tiene diabetes (1) o no (0). Las principales características del conjunto de datos son el número de embarazos, la concentración de glucosa en sangre, la presión arterial diastólica, el espesor del pliegue cutáneo del tríceps, el nivel de insulina, el índice de masa corporal (IMC), la función del pedigrí de diabetes y la edad.

2) Exploración inicial del conjunto de datos: Una vez cargados los datos, se realizó una exploración inicial para comprender la naturaleza del conjunto de datos y detectar posibles problemas. Se utilizó la función `info()` de Pandas para obtener información sobre el tipo de cada columna, la cantidad de valores no nulos y la presencia de valores faltantes. Además, se utilizó la función `describe()` obtener una descripción estadística de las variables, lo cual permitió identificar outliers y observar las distribuciones de cada variable.

Adicionalmente, se revisaron los valores únicos por columna para detectar posibles errores o anomalías en los datos. Esta exploración inicial fue crucial para entender la naturaleza de las variables biométricas y evaluar si había problemas como datos incompletos, distribuciones inusuales o valores atípicos.

3) Manejo de valores faltantes: Durante la exploración inicial, se detectaron valores de 0 en variables donde estos no eran fisiológicamente posibles, como en los niveles de glucosa, presión arterial, espesor del pliegue cutáneo, insulina y IMC. Estos valores se consideraron como datos faltantes y fueron reemplazados por `NaN` para tratarlos adecuadamente en los pasos de preprocesamiento posteriores.

Una vez identificados los valores faltantes, se procedió a imputarlos utilizando métodos estadísticos. Para la variable Glucose, se utilizó la media, dado que no presentaba tantos valores extremos o outliers. Para las otras variables, como BloodPressure, SkinThickness, Insulin y BMI, se utilizó la mediana, ya que estas variables presentaban distribuciones más dispersas y con presencia de valores atípicos.

4) Detección y Tratamiento de Outliers: La presencia de outliers puede afectar el rendimiento de los modelos de Machine Learning, especialmente aquellos basados en distancia, como KNN. Para detectar y tratar estos valores atípicos, se siguieron los siguientes pasos:

Identificación de outliers: Se utilizaron diagramas de caja (boxplots) para visualizar la dispersión de los datos y detectar outliers en las columnas que contenían valores numéricos. Este método gráfico es efectivo para identificar valores que se encuentran fuera del rango intercuartílico (IQR).

Método del rango intercuartílico (IQR): Para cuantificar los outliers, se calculó el IQR, definido como la diferencia entre el tercer cuartil (Q3) y el primer cuartil (Q1). Los valores fuera del rango estándar de 1.5 veces el IQR fueron considerados outliers:

$$\text{Outliers} = Q1 - 1.5 \times \text{IQR} \quad \text{o} \quad Q3 + 1.5 \times \text{IQR}$$

Imputación de outliers: En lugar de eliminar los outliers, se optó por imputar estos valores con la mediana de la columna correspondiente, una técnica robusta ante valores extremos. La eliminación de outliers hubiera reducido aún más el tamaño del conjunto de datos, mientras que la imputación permitió mantener el máximo de información posible.

5) Exploración detallada de características: Luego de hacer la imputación de datos, se llevó a cabo un análisis exhaustivo de las características del conjunto de datos mediante visualizaciones gráficas para observar las distribuciones individuales de cada variable y las relaciones entre ellas. Para cada variable, se generaron histogramas con KDE (Kernel Density Estimation) que permitieron visualizar la forma de las distribuciones. Adicionalmente, se utilizaron diagramas de caja (boxplots)

para detectar la presencia de valores atípicos que pudieran influir negativamente en los modelos.

También se generó un mapa de calor para visualizar las correlaciones entre las variables, lo cual permitió identificar qué características estaban más fuertemente asociadas con la variable objetivo Outcome. Este paso fue importante para entender qué variables tenían mayor influencia en la predicción de la diabetes.

6) Evaluación del balance de clases: Seguidamente, era necesario verificar el balance de clases antes de entrenar los modelos, ya que un conjunto de datos desbalanceado puede generar modelos que tienen dificultades para predecir correctamente la clase minoritaria. Para visualizar el desbalance, se utilizó un gráfico de barras para visualizar la distribución de las clases, la cual demostró que la variable objetivo Outcome tenía una distribución desequilibrada (aproximadamente 65% de pacientes sin diabetes y 35% con diabetes),

Este desbalance justificó el uso de la técnica SMOTE (Synthetic Minority Over-sampling Technique), que fue aplicada para generar ejemplos sintéticos de la clase minoritaria (pacientes con diabetes). Esta técnica ayudó a balancear el conjunto de datos y mejorar el rendimiento de los modelos, especialmente en términos de recall, que mide la capacidad del modelo para identificar correctamente los casos positivos (pacientes con diabetes).

```
smote = SMOTE(random_state=42)
X_resampled, y_resampled =
    smote.fit_resample(X_train, y_train)
```

7) División del conjunto de datos: El conjunto de datos fue dividido en tres subconjuntos: entrenamiento (70%), validación (15%) y prueba (15%). La división se realizó de manera estratificada para garantizar que la proporción de clases en cada subconjunto reflejara la distribución original del conjunto de datos. Esta estratificación fue esencial para asegurar que los modelos fueran entrenados y evaluados en condiciones comparables, evitando que un subconjunto tuviera un desbalance diferente al conjunto original.

8) Normalización de las características: Dado que el algoritmo de K-Nearest Neighbors (KNN) utiliza medidas de distancia para hacer predicciones, es fundamental que todas las características estén en la misma escala. Características con rangos de valores más amplios podrían dominar las predicciones. Para evitar esto, se aplicó la técnica de normalización mediante MinMaxScaler, que transforma todas las características para que se encuentren en el rango de [0, 1]. Esta técnica es especialmente útil para reducir el impacto de los outliers y asegurar que todas las variables contribuyan de manera equitativa al modelo.

```
scaler = MinMaxScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_val_scaled = scaler.transform(X_val)
X_test_scaled = scaler.transform(X_test)
```

9) Implementación de los modelos de clasificación:

a) Regresión Logística: La regresión logística es un modelo lineal de clasificación que estima la probabilidad de que una instancia pertenezca a una clase determinada. Este modelo es adecuado para conjuntos de datos donde las relaciones entre las variables son aproximadamente lineales. Para abordar el desbalance de clases y mejorar la capacidad del modelo para predecir correctamente la clase minoritaria (diabetes), se utilizó la opción `class_weight='balanced'`, que ajusta automáticamente los pesos de las clases en función de su frecuencia en el conjunto de datos.

Con el objetivo de optimizar el rendimiento del modelo, se entrenaron tres versiones de regresión logística utilizando diferentes conjuntos de hiperparámetros, para determinar el conjunto que ofrecía el mejor rendimiento en términos de *accuracy*, *precision*, *recall*, *F1-score*, y *AUC-ROC*.

Hiperparámetros utilizados

Set 1: En este primer experimento, se utilizaron los hiperparámetros predeterminados del modelo de regresión logística. El modelo utilizó:

- Penalización: L2 (Regularización de Ridge)
- C: 1.0 (Valor predeterminado de regularización)
- Solver: `lbfgs` (Solver estándar para problemas de regresión logística)
- `Max_iter`: 1000 (Número de iteraciones para asegurar convergencia)

Este set sirvió como referencia base para los modelos más avanzados y permitió evaluar la necesidad de ajustar los parámetros para mejorar el rendimiento.

Set 2: En el segundo experimento, se aplicaron cambios al valor de `C` (parámetro de regularización) y al solver utilizado para mejorar la convergencia y ajustar la magnitud de la regularización. Los hiperparámetros ajustados fueron:

- Penalización: L2
- C: 0.1 (Aumento en la regularización, lo que tiende a reducir el riesgo de sobreajuste)
- Solver: `lbfgs`
- `Max_iter`: 1000

Este set permitió explorar un ajuste de regularización que redujera el riesgo de sobreajuste, especialmente en un conjunto de datos con muchas características predictoras y potenciales interacciones.

Set 3: Para el tercer experimento, se introdujo la regularización ElasticNet, que combina las penalizaciones L1 (Lasso) y L2 (Ridge), y puede ser efectiva en la selección de características y para reducir el sobreajuste en datos con correlaciones complejas entre variables. Los hiperparámetros utilizados fueron:

- Penalización: ElasticNet (Combinación de L1 y L2)
- C: 0.01 (Mayor regularización)
- Solver: `saga` (Compatibilidad con ElasticNet)
- `l1_ratio`: 0.5 (Equilibrio entre penalización L1 y L2)
- `Max_iter`: 1500 (Número de iteraciones más alto para asegurar la convergencia)

Este set permitió evaluar el impacto de una penalización mixta (L1 y L2) en el modelo, optimizando tanto la selección de características como el ajuste general del modelo a los datos.

Los modelos fueron evaluados tanto en el conjunto de validación como en el conjunto de prueba, y los resultados se compararon para determinar cuál de los tres sets de hiperparámetros proporcionaba el mejor rendimiento, considerando métricas clave como *AUC-ROC* y *F1-score*.

b) K-Nearest Neighbors (KNN): El segundo algoritmo utilizado en este proyecto fue K-Nearest Neighbors (KNN). KNN es un algoritmo basado en instancias que clasifica una nueva muestra asignándole la clase predominante de sus vecinos más cercanos. Al no realizar suposiciones sobre la distribución subyacente de los datos, KNN puede ser particularmente efectivo en problemas de clasificación no lineales. Sin embargo, su rendimiento depende en gran medida del número de vecinos (k) y de la escala de las características.

Uno de los desafíos clave en la implementación de KNN es seleccionar el valor óptimo para k, el número de vecinos considerados en el proceso de clasificación. Un valor bajo de k puede hacer que el modelo sea demasiado sensible al ruido y a los outliers, mientras que un valor muy alto puede suavizar excesivamente el modelo, perdiendo detalles importantes.

Para encontrar el valor óptimo de k, se llevó a cabo una búsqueda exhaustiva mediante la variación de k desde 1 hasta 250. Para cada valor de k, el modelo fue entrenado utilizando el conjunto de datos sobremuestreado con SMOTE, y se evaluó su rendimiento en el conjunto de validación utilizando las siguientes métricas: *Accuracy*, *Precision*, *Recall*, *F1-Score* y *AUC-ROC*.

Luego, en lugar de seleccionar un único valor de k basado en una sola métrica, se analizaron las cinco métricas clave para obtener una visión integral del rendimiento del modelo:

- **Accuracy:** Proporción de predicciones correctas.
- **Precision:** Porcentaje de verdaderos positivos sobre todas las predicciones positivas.
- **Recall:** Proporción de verdaderos positivos correctamente identificados.
- **F1-Score:** Media armónica entre *precision* y *recall*, una métrica útil en conjuntos de datos desbalanceados.
- **AUC-ROC:** Área bajo la curva ROC, que mide la capacidad del modelo para distinguir entre las clases positiva y negativa.

Se seleccionaron los mejores valores de k basados en cada una de estas métricas. Esto permitió identificar un k óptimo para cada objetivo de rendimiento, en lugar de depender de una única métrica.

Las matrices de confusión también fueron generadas para visualizar los resultados de la clasificación en términos de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos.

```
conf_matrix = confusion_matrix(y_test, y_pred)
ConfusionMatrixDisplay(conf_matrix).plot(cmap='Blues')
plt.show()
```

10) Interpretación y comparación de resultados: Finalmente, los resultados obtenidos de ambos modelos fueron comparados utilizando las métricas de rendimiento mencionadas anteriormente. Se identificó cuál de los dos modelos daba mejores resultados con las diferentes métricas, y se determinó cuál era más adecuado para predecir la diabetes en este conjunto de datos.

B. *Anemia Dataset*

1) Carga del conjunto de datos: Cargamos de igual manera el conjunto de datos utilizando Pandas, con esto obtenemos el dataframe listo para manipular y explorar el archivo con mayor facilidad. Este conjunto de datos cuenta con 1421 registros, en el cual cada uno representa un paciente de género hombre o mujer, además está compuesto por seis columnas, las cuales aportan información clave para el análisis de esta enfermedad [5].

2) Exploración inicial del conjunto de datos: El uso de `info()` y `describe()` permitieron obtener un vista general de la información que ofrece el conjunto de datos, además de una descripción estadística que contiene: promedio, desviación estándar, mínimo, primer cuartil (25%), segundo cuartil (50%), tercer cuartil (75%) y máximo, los cuales nos ayudó a comprender la variabilidad de los datos, identificar posibles valores atípicos en el análisis de la anemia y a tener una noción del dataset.

Se realizaron otras exploraciones y revisiones en el conjunto de datos como: contar el número de hombres y mujeres, calcular cuántos valores únicos existen por cada columna y calcular cuántos valores nulos existen por columna.

3) Exploración detallada de características: Procedimos a analizar y comparar características hematológicas de ambos géneros en el conjunto de datos, para esto visualizamos las distribuciones individuales de cada característica mediante histogramas con Kernel Density Estimation (KDE) para visualizar la distribución de sus datos e identificar posibles valores atípicos o clústeres, y también mediante un mapa de calor que presenta las correlaciones entre variables con el fin de detectar relaciones importantes entre las características. Igualmente, se exploraron las relaciones entre múltiples variables mediante el uso de pairplot y relaciones entre dos variables con la ayuda de scatter plot.

4) Detección de outliers: Los outliers son valores extremos que se desvían significativamente de la tendencia general de un conjunto de datos. Estos puntos son atípicos y se encuentran alejados de la mayoría de las observaciones [8]. Por ende, seleccionamos las columnas relevantes para detectar posibles outliers: Gender, Hemoglobin, MCH, MCHC y MCV. Luego, se utilizaron los diagramas de caja para visualizar estos outliers basados en el método del rango intercuartílico, de la misma manera implementación realizada con el dataset de Diabetes.

5) Evaluación del balance de clases: Se realizó una evaluación del balance de clases de la columna de Result, con el fin de identificar posibles desbalances y revisar si puede llegar a afectar la capacidad de los modelos para generalizar correctamente. Se identificó el porcentaje para los pacientes que no tiene anemia (0) y los que sí tienen anemia. Debido a esto, utilizamos el sobremuestreo usando SMOTE ya que existe un desbalance no tan grande, pero que de igual forma puede afectar el desempeño de los modelos. Esta técnica genera ejemplos

sintéticos de la clase minoritaria (pacientes con anemia), a pesar de ser un dataset relativamente de gran tamaño. Se optó por aplicar sobremuestreo en lugar de submuestreo, dado que el objetivo es preservar la mayor cantidad de datos posible y evitar la eliminación de información valiosa que podría ocurrir con el submuestreo al balancear las clases.

6) División del conjunto de datos: Se utilizó una proporción de 70% de entrenamiento, 15% de validación y 15% de prueba. El conjunto de entrenamiento se emplea para ajustar el modelo y aprender los patrones presentes en los datos. La validación permite ajustar los hiperparámetros y evitar sobreajuste, mientras que el conjunto de prueba se utiliza exclusivamente para evaluar el rendimiento del modelo de forma objetiva, asegurando que no haya sido influido durante el proceso de entrenamiento o ajuste. Esta división garantiza una evaluación adecuada del modelo en datos no vistos.

7) Normalización y escalabilidad: Dado que algunas variables tienen escalas de magnitud distintas, era necesario transformar los datos a una escala común. Esto se logró utilizando una normalización MinMaxScaler, que ajusta los valores dentro del rango de 0 a 1 antes de entrenar el modelo, e implementando la prueba de Shapiro-Wilk par saber si los datos siguen una distribución normal.

8) Implementación de los modelos de clasificación:

a) Regresión Logística: Es una técnica de análisis de datos que utiliza las matemáticas para encontrar las relaciones entre dos factores de datos. Luego, utiliza esta relación para predecir el valor de uno de esos factores basándose en el otro. La predicción final consiste en una clasificación binaria, solo puede tener dos resultados [6].

Se utilizó la clase `LogisticRegression` de la librería `sklearn`, una herramienta ampliamente reconocida por su eficiencia en la construcción de modelos de clasificación. El modelo fue entrenado con las características normalizadas del conjunto de datos, donde aplicamos `class_weight='balanced'` para ajustar los pesos de las clases, mejorando el rendimiento en la clase minoritaria. Comparamos dicho modelo con los datos de sobremuestreo y datos escalados, para confirmar si el sobremuestreo de clases tiene ventaja sobre los datos no sobremuestrados.

Seguidamente, se definió una cuadrícula de posibles valores de hiperparámetros incluyendo el parámetro de regularización C con valores 0.01, 0.1, 1; el tipo de penalización con l1 y l2; y el solver liblinear, newton-cg, lbfgs, saga. Cada combinación fue fue evaluada para identificar la más óptima, las cuales fueron entrenada con el fin de calcular las métricas importantes y observar su rendimiento.

b) K-Nearest Neighbors (KNN): Se llevó a cabo una evaluación exhaustiva del modelo K-Nearest Neighbors (KNN) para determinar el número óptimo de vecinos (n_neighbors) que maximice su rendimiento. Para ello, se entrenó el modelo utilizando una variación en el número de vecinos, considerando un rango de 1 a 100, con los datos balanceados previamente mediante técnicas de sobremuestreo. Esto permitió que el modelo tomara en cuenta de manera adecuada el equilibrio de clases en el conjunto de entrenamiento.

Una vez entrenado el modelo con cada valor de n_neighbors, se procedió a evaluarlo en el conjunto de validación. Luego, se representaron gráficamente las métricas obtenidas en función del número de vecinos, lo cual permitió visualizar el comportamiento del modelo y facilitar la identificación del valor óptimo en términos de desempeño general.

Posteriormente, se entrenó el modelo KNN utilizando el número de vecinos que optimizó el AUC-ROC, al ser esta métrica una de las más importantes en la clasificación binaria. Se evaluó su desempeño en el conjunto de prueba, calculando nuevamente las métricas para verificar su rendimiento final. Por último se generó la curva ROC para el modelo basado en el mejor valor de AUC-ROC, lo cual permitió visualizar gráficamente el desempeño del modelo

9) Interpretación y comparación de datos: En este apartado, se llevó a cabo una comparación entre dos modelos de clasificación: Regresión Logística y K-Nearest Neighbors (KNN), utilizando la curva ROC como herramienta principal de evaluación. Se calculó la curva ROC acompañada de su respectivo valor de AUC-ROC, el cual indica el área bajo la curva y es una medida global del desempeño del modelo.

Se graficaron ambas curvas ROC para comparar visualmente el rendimiento de los dos modelos en términos de su capacidad de discriminación. Esta comparación proporciona información valiosa para poder identificar cuál de los dos modelos presenta un mejor balance entre la tasa de verdaderos y falsos positivos.

III. Resultados

A. Diabetes Dataset

1) Carga de datos y exploración inicial: Como se mencionó en la sección de la metodología, se cargaron los datos del archivo `diabetes.csv`, que contiene 768 registros y 9 columnas. Los datos incluyen variables como la cantidad de embarazos (*Pregnancies*), nivel de glucosa (*Glucose*), presión arterial diastólica (*BloodPressure*), grosor de la piel del tríceps (*SkinThickness*), insulina sérica (*Insulin*), índice de masa corporal (*BMI*), función del pedigrí de la diabetes (*DiabetesPedigreeFunction*), edad (*Age*), y la variable objetivo (*Outcome*), que indica si el paciente tiene o no diabetes.

El análisis reveló valores anómalos, particularmente en las columnas fisiológicas donde no se espera que los valores sean 0, como *Glucose*, *BloodPressure*, *SkinThickness*, *Insulin* y *BMI*. En la revisión inicial, se observaron varios ceros en estas columnas, lo que sugirió la necesidad de manejo de valores faltantes.

2) Identificación y manejo de valores faltantes: Se identificaron las columnas con valores de 0 en las variables fisiológicas clave, las cuales se reemplazaron con valores `NaN` para tratarlos como valores faltantes. En la tabla inicial, se detectaron los siguientes valores faltantes:

- *Glucose*: 5 valores faltantes
- *BloodPressure*: 35 valores faltantes
- *SkinThickness*: 227 valores faltantes
- *Insulin*: 374 valores faltantes
- *BMI*: 11 valores faltantes

Para manejar los valores faltantes, se decidió utilizar la media para imputar los valores faltantes de *Glucose*, ya que su distribución no mostró muchos valores extremos (outliers). Por otro lado, para las demás variables (*BloodPressure*, *SkinThickness*, *Insulin*, y *BMI*), se utilizó la mediana para evitar el sesgo causado por los valores extremos.

Después de la imputación, se verificó que no quedaran valores faltantes en el conjunto de datos.

3) Distribuciones individuales de características: Se realizaron histogramas con *KDE* para cada variable con el fin de explorar visualmente la distribución de los datos. En general, se observó que varias variables mostraban distribuciones no normales y, en algunos casos, una gran dispersión. Por ejemplo:

- *Pregnancies* tiene una distribución sesgada a la derecha, lo que indica que la mayoría de las pacientes han tenido pocos embarazos(Figura. 1).

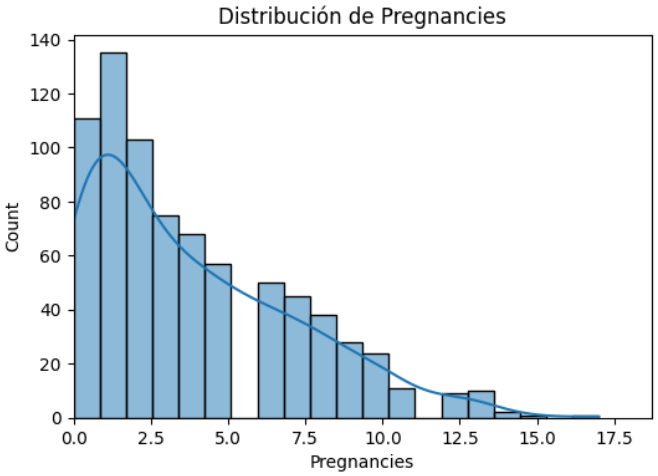

Fig. 1. Distribución de Embarazos

- *Glucose* muestra una distribución más centrada, pero con algunos valores extremos altos
- *Insulin* tiene una alta concentración de valores bajos, lo cual refleja la presencia de muchos valores de insulina considerados como faltantes (que se imputaron)

4) Análisis de correlaciones: Además, se generó un mapa de calor (Figura. 2) para visualizar la correlación entre las variables del conjunto de datos. Las variables más fuertemente correlacionadas con la variable objetivo (*Outcome*) fueron:

- *Glucose*: fuerte correlación positiva con la diabetes.
- *BMI* y *Age*: también mostraron una correlación positiva, aunque más moderada.

Este análisis sugiere que la glucosa es una característica clave para predecir la diabetes, lo cual es coherente con lo que se espera clínicamente.

5) Balance de clases: El conjunto de datos está desbalanceado, con un 65% de registros de pacientes sin diabetes (*Outcome*=0) y solo un 35% con diabetes (*Outcome*=1) (Figura. 3). Esto requiere la implementación de técnicas para manejar el desbalance, como la aplicación del método de sobremuestreo *SMOTE*.

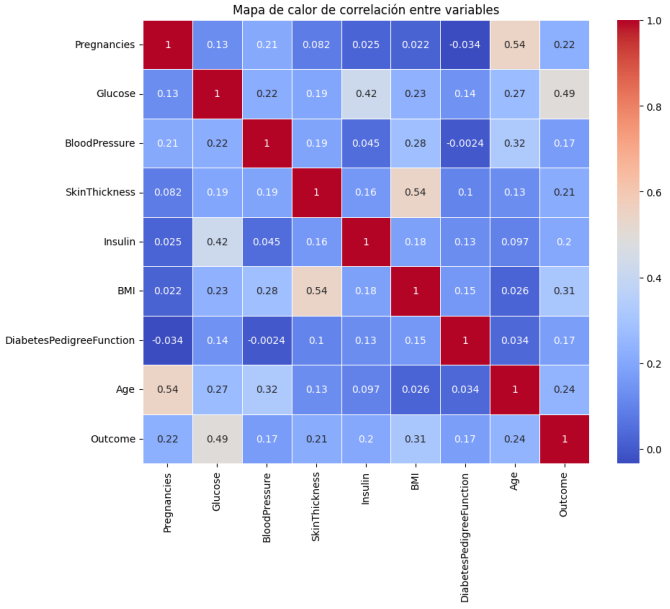

Fig. 2. Mapa de calor de correlación entre variables Diabetes Dataset

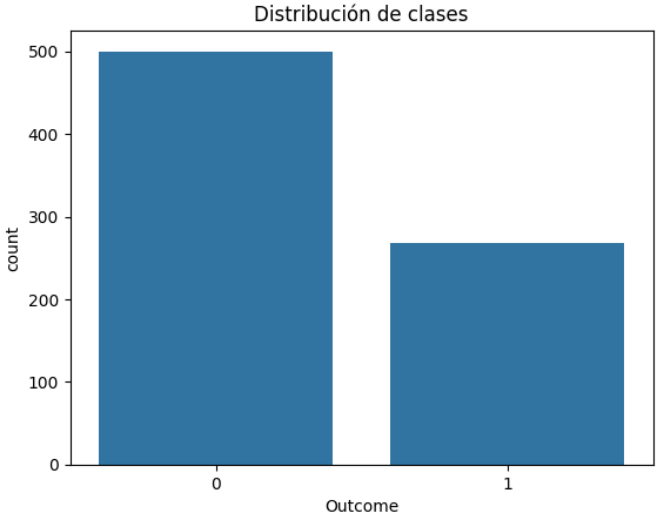

Fig. 3. Distribución de Clases Diabetes Dataset

6) Detección y manejo de outliers: Se utilizaron diagramas de caja para detectar outliers en las variables fisiológicas. La mayoría de las columnas presentaron outliers notables, especialmente *Insulin*, con un 45% de valores considerados atípicos (Figura. 4). Para mitigar el impacto de estos valores extremos, se reemplazaron por la mediana en lugar de eliminarlos, evitando así reducir el tamaño del conjunto de datos. En al Figura 5 se muestra el diagrama de caja en Insulin después de aplicar el manejo de outliers.

7) Resultados del análisis de pares de características: En la Figura 6, se muestra el análisis de pares de características entre *Age*, *Glucose*, *BMI* e *Insulin*, en relación con la variable objetivo *Outcome* (0: no tiene diabetes, 1: tiene diabetes). Los

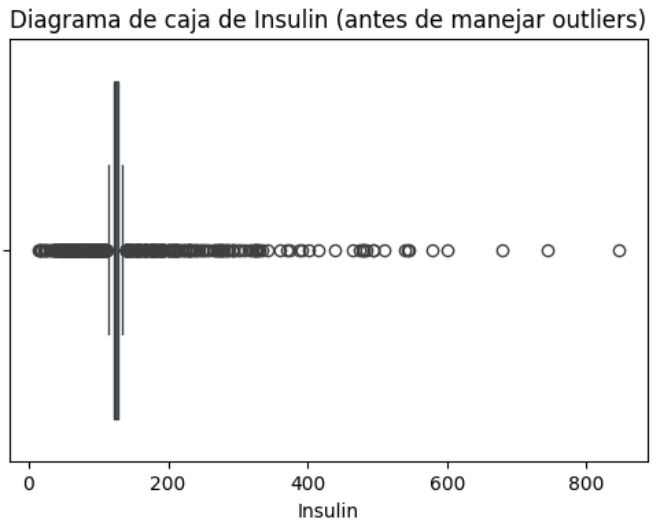

Fig. 4. Diagrama de caja de Insulina (antes de manejar outliers)

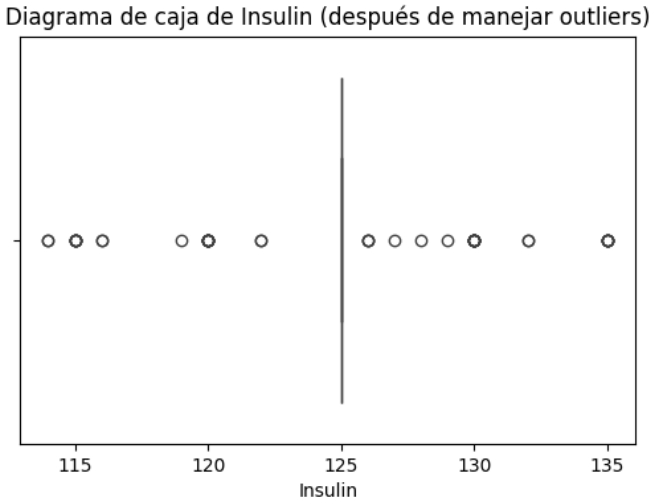

Fig. 5. Diagrama de caja de Insulina (después de manejar outliers)

puntos naranjas representan a los pacientes con diabetes, y los puntos azules, a los pacientes sin diabetes.

a) Distribuciones individuales: Las distribuciones de las variables indican tendencias importantes:

- **Edad (Age)**: La mayoría de los pacientes sin diabetes están en el rango de edad más joven (20 a 40 años), mientras que los pacientes con diabetes tienden a estar más distribuidos en un rango amplio de edades, concentrándose a partir de los 40 años.
- **Glucosa (Glucose)**: Existe una clara separación entre los pacientes con diabetes, que tienden a tener niveles de glucosa más altos. Los pacientes sin diabetes presentan una distribución de glucosa más centralizada.
- **IMC (BMI)**: Aunque las diferencias no son tan pronunciadas como en la glucosa, los pacientes con diabetes tienden a tener un IMC más elevado (superior a 30), lo que confirma su relación con el riesgo de diabetes.
- **Insulina (Insulin)**: La variable insulina presenta poca variabilidad en los niveles y no parece ser un buen predictor en este análisis, lo que podría estar influenciado por los valores imputados en el preprocesamiento.

b) Relaciones entre variables: Las relaciones entre pares de variables también revelan patrones clave:

- **Glucosa y Edad (Glucose vs. Age)**: Los pacientes con niveles altos de glucosa (>120) y mayor edad tienden a tener diabetes, aunque la relación no es lineal.
- **IMC y Glucosa (BMI vs. Glucose)**: Existe una correlación entre IMC alto y glucosa elevada en pacientes con diabetes, sugiriendo que el sobrepeso es un factor importante en el desarrollo de la enfermedad.
- **Insulina y Glucosa (Insulin vs. Glucose)**: No se observa una separación clara entre pacientes con y sin diabetes en esta relación, probablemente debido a la imputación de valores en la variable insulina.

Este análisis revela que las variables **Glucose** y **BMI** son las más relevantes para diferenciar entre pacientes con y sin diabetes, con un enfoque particular en la glucosa, que muestra la separación más clara. La variable **Age**, aunque menos influyente por sí sola, también contribuye en combinación con otros factores.

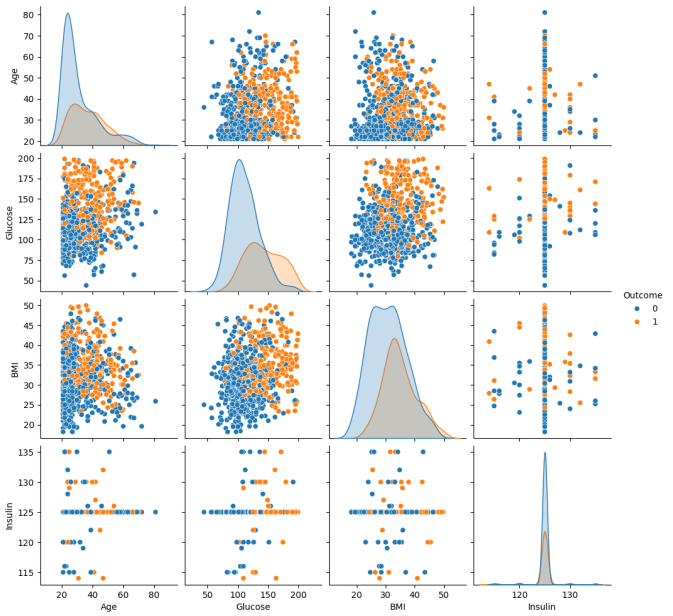

Fig. 6. Gráfico de pares de características. Las variables analizadas son: Age, Glucose, BMI e Insulin, en relación con el Outcome (0: no diabetes, 1: diabetes).

8) Prueba de normalidad y escalado de características: Luego, se aplicó la prueba de *Shapiro-Wilk* para verificar si las variables seguían una distribución normal. Los resultados mostraron que la mayoría de las características no seguían una distribución normal (p ¡ 0.05). Como resultado, se utilizó la técnica de escalado *MinMaxScaler* para normalizar las variables,

ya que es más adecuado para algoritmos como *KNN* que se basan en la distancia.

9) Sobremuestreo con SMOTE: Además, dado el desbalance de clases, se utilizó el método de sobremuestreo sintético (*SMOTE*) para equilibrar el conjunto de entrenamiento. Después de aplicar *SMOTE*, se consiguió un balance perfecto entre las clases, con un 50% de casos positivos y 50% de negativos. Y además, al visualizar las distribuciones antes y después de aplicar SMOTE, se verificó que las distribuciones de las columnas antes y después de aplicar el método no se alteraron significativamente, como se puede ver en el caso de la distribución de Glucose en las Figuras 7 y 8.

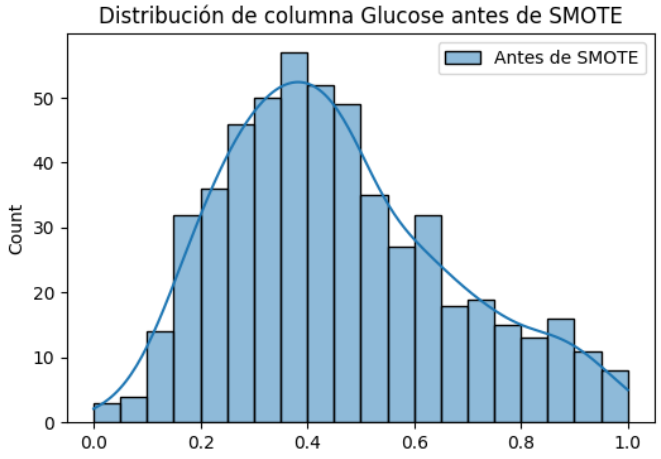

Fig. 7. Distribución de columna Glucose antes de SMOTE

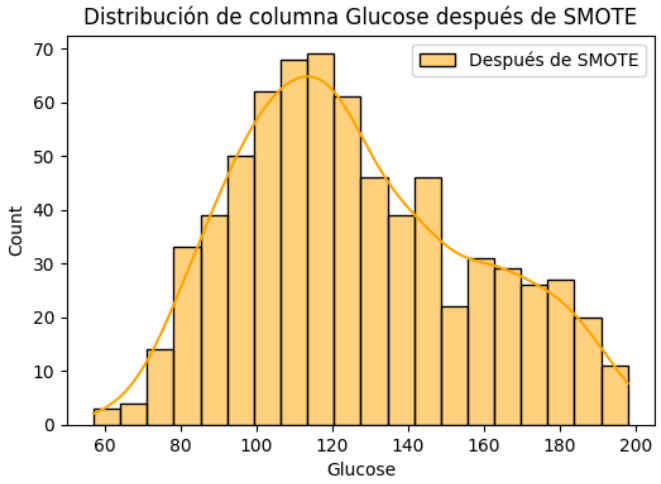

Fig. 8. Distribución de columna Glucose después de SMOTE

B. Resultados del entrenamiento y evaluación de modelos

1) Entrenamiento y evaluación de Regresión Logística: Siguiendo la metodología, se entrenaron tres versiones del modelo de regresión logística, variando los hiperparámetros para evaluar su impacto en las métricas de rendimiento: *accuracy*, *precision*, *recall*, *F1-score*, y *AUC-ROC*. A continuación, se detallan los resultados obtenidos para cada conjunto de hiperparámetros.

a) Set 1 de Hiperparámetros: El primer modelo utilizó los hiperparámetros predeterminados de la regresión logística: penalización L2 (regularización Ridge), valor de C=1.0, solver `lbfgs` y un máximo de 1000 iteraciones. Este set es considerado un punto de referencia para evaluar el impacto de las modificaciones posteriores. Los resultados para el conjunto de validación fueron los siguientes:

- **Accuracy**: 0.7043
- **Precision**: 0.5577
- **Recall**: 0.7250
- **F1-Score**: 0.6304
- **AUC-ROC**: 0.8090

El modelo demostró una buena capacidad para identificar correctamente los casos positivos (*recall* de 72.50%), a pesar de que la precisión fue moderada (55.77%), lo que indica una tasa relativamente alta de falsos positivos. El área bajo la curva *ROC* (*AUC-ROC*) de 0.8090 indica una capacidad discriminativa adecuada entre las clases.

En el conjunto de prueba, el rendimiento del modelo mejoró:

- **Accuracy**: 0.7931
- **Precision**: 0.7179
- **Recall**: 0.6829
- **F1-Score**: 0.7000
- **AUC-ROC**: 0.8670

El aumento en precisión (71.79%) y *accuracy* (79.31%) en el conjunto de prueba sugiere que el modelo fue capaz de generalizar bien a nuevos datos. Además, el *AUC-ROC* de 0.8670 confirma que el modelo mejoró su capacidad para discriminar entre pacientes con y sin diabetes en el conjunto de prueba.

La matriz de confusión (Figura. 9) muestra que, aunque el modelo identificó correctamente muchos casos positivos, aún existieron algunos errores en la clasificación de casos negativos como positivos. La aplicación de class_weight='balanced' y el uso de *SMOTE* contribuyeron a mejorar el rendimiento en un conjunto de datos desbalanceado.

b) Set 2 de Hiperparámetros: Para el segundo set, se ajustó el valor de C a 0.1 (aumentando la regularización para evitar el sobreajuste), mientras se mantuvo la penalización L2, el solver `lbfgs` y un máximo de 1000 iteraciones. Este set tenía como objetivo reducir la varianza del modelo y mejorar la estabilidad.

Los resultados para el conjunto de validación fueron:

- **Accuracy**: 0.6957
- **Precision**: 0.5490
- **Recall**: 0.7000
- **F1-Score**: 0.6154
- **AUC-ROC**: 0.8027

Este modelo mostró una ligera mejora en el *recall*, pero disminuyó en precisión (54.90%), lo que implica un aumento de falsos positivos. Sin embargo, el rendimiento general fue comparable al Set 1, con un *AUC-ROC* de 0.8027, lo que indica

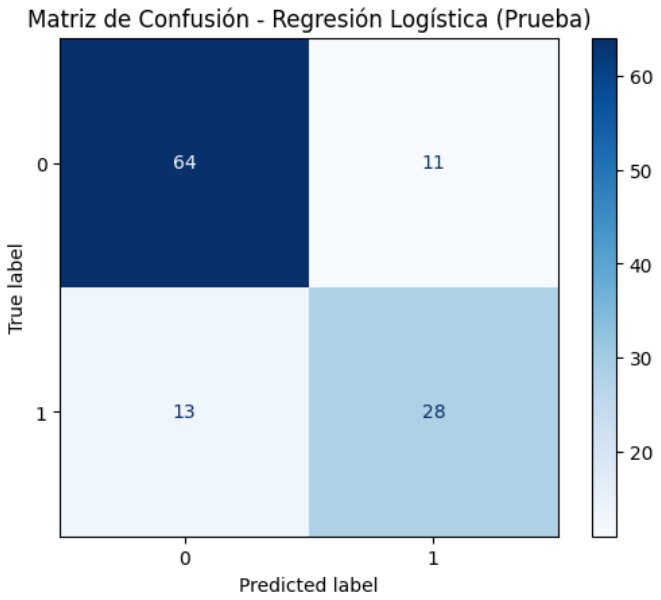

Fig. 9. Matriz de Confusión Set 1 Regresión Logística (Prueba)

que el ajuste no afectó de manera significativa la capacidad del modelo para discriminar entre clases.

En el conjunto de prueba, el rendimiento fue consistente:

- **Accuracy**: 0.7759
- **Precision**: 0.6829
- **Recall**: 0.6829
- **F1-Score**: 0.6829
- **AUC-ROC**: 0.8686

La similitud entre los valores de *precision* y *recall* muestra un mejor balance entre ambos, y el *AUC-ROC* de 0.8686 en el conjunto de prueba es ligeramente superior al del Set 1, indicando que el modelo mantiene una alta capacidad discriminativa.

c) Set 3 de Hiperparámetros: El tercer set utilizó una combinación de penalización L1 y L2 mediante ElasticNet (con C=0.01, solver `saga` y l1-ratio=0.5). Este ajuste buscaba mejorar la selección de características al mismo tiempo que reducía el riesgo de sobreajuste.

En el conjunto de validación, se observaron los siguientes resultados:

- **Accuracy**: 0.6870
- **Precision**: 0.5385
- **Recall**: 0.7000
- **F1-Score**: 0.6087
- **AUC-ROC**: 0.7987

Si bien el modelo mantuvo un buen *recall* (70%), los resultados globales fueron levemente inferiores a los de los sets anteriores, particularmente en precisión (53.85%).

En el conjunto de prueba, los resultados mejoraron:

- **Accuracy**: 0.7931
- **Precision**: 0.7073
- **Recall**: 0.7073
- **F1-Score**: 0.7073
- **AUC-ROC**: 0.8728

El modelo mostró un buen equilibrio entre precisión y *recall* en el conjunto de prueba, lo que se refleja en un *F1-Score* de 0.7073. El valor de *AUC-ROC* de 0.8728 también fue el más alto entre los tres sets, lo que sugiere que ElasticNet podría ofrecer una ventaja al combinar ambas formas de regularización, particularmente cuando se busca minimizar tanto el sobreajuste como los errores de clasificación.

2) Resultados del entrenamiento y evaluación de modelos K-Nearest Neighbors (KNN): El modelo *K-Nearest Neighbors* (KNN) fue evaluado variando el número de vecinos (k) para observar su impacto en las métricas de rendimiento, tales como *accuracy*, *precision*, *recall*, *F1-Score*, y *AUC-ROC*. Se probaron valores de k desde 1 hasta 150, y se seleccionaron los mejores valores para cada métrica en función de los resultados obtenidos en los conjuntos de validación y prueba.

La Figura 10 muestra el rendimiento del modelo *K-Nearest Neighbors* (KNN) al variar el número de vecinos (k) desde 1 hasta 150, evaluando las métricas de *Accuracy*, *Precision*, *Recall*, *F1-Score* y *AUC-ROC*. Se observa que las métricas experimentan fluctuaciones significativas cuando k es pequeño, especialmente en *Precision* y *Recall*, estabilizándose a partir de $k \approx 20$. El *AUC-ROC* se mantiene alto y estable alrededor de 0.85, indicando una buena capacidad del modelo para discriminar entre clases. Los valores óptimos de k se encuentran entre 20 y 30, proporcionando un equilibrio sólido entre las métricas, lo que sugiere que el modelo tiene un buen rendimiento general para la clasificación.

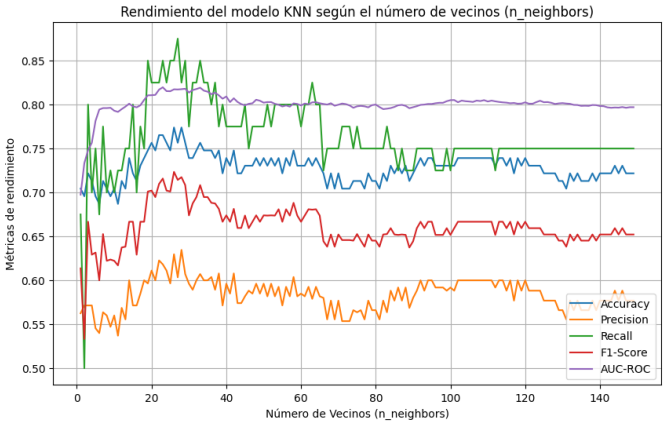

Fig. 10. Rendimiento del modelo KNN según el número de vecinos

a) Selección del mejor valor de k para cada métrica: Después de evaluar diferentes valores de k, se encontraron los siguientes resultados en el conjunto de validación:

- Mejor valor de k basado en *Accuracy*: $k = 26$
- Mejor valor de k basado en *Precision*: $k = 28$
- Mejor valor de k basado en *Recall*: $k = 27$
- Mejor valor de k basado en *F1-Score*: $k = 26$

Estos valores representan los mejores resultados alcanzados para cada métrica, permitiendo seleccionar el modelo adecuado

según el objetivo de optimización, ya sea minimizar los falsos positivos (*Precision*) o maximizar la detección de la clase positiva (*Recall*).

b) Evaluación en el conjunto de prueba: **1. Mejor modelo basado en *Accuracy* (*k* = 26)**

- **Accuracy**: 0.7586
- **Precision**: 0.6327
- **Recall**: 0.7561
- **F1-Score**: 0.6889

El modelo basado en *Accuracy* mostró un buen equilibrio entre las diferentes métricas, destacando una alta capacidad para identificar correctamente tanto los casos positivos como negativos, sin sacrificar demasiada precisión. La matriz de confusión (Figura. 11) indicó 18 falsos positivos y 10 falsos negativos, lo que refleja un equilibrio razonable entre precisión y *recall*.

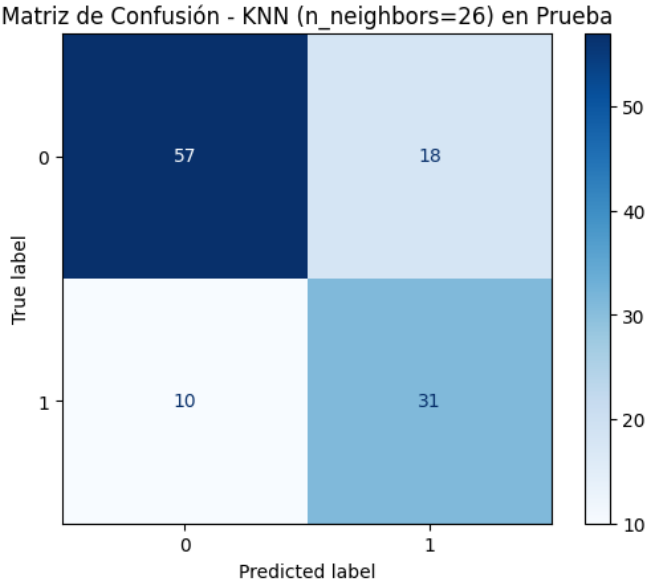

Fig. 11. Matriz de Confusión Accuracy

2. Mejor modelo basado en *Precision* (*k* = 28)

- **Accuracy**: 0.7672
- **Precision**: 0.6458
- **Recall**: 0.7561
- **F1-Score**: 0.6966

El modelo basado en *Precision* fue diseñado para minimizar los falsos positivos. Como resultado, logra un buen desempeño en la precisión (64.58%), lo que es crucial en aplicaciones donde es preferible minimizar los falsos positivos, como en el diagnóstico médico para evitar diagnósticos erróneos de diabetes en pacientes sanos. La matriz de confusión mostró 17 falsos positivos y 10 falsos negativos.

3. Mejor modelo basado en *Recall* (*k* = 27)

- **Accuracy**: 0.7655
- **Precision**: 0.6400
- **Recall**: 0.7800
- **F1-Score**: 0.7037

El modelo basado en *Recall* fue optimizado para maximizar la capacidad de detectar correctamente los casos positivos (78% de *recall*), lo que es especialmente importante en problemas como la detección de enfermedades, donde se prioriza evitar falsos negativos. La matriz de confusión reflejó 19 falsos positivos y 10 falsos negativos. **4. Mejor modelo basado en *F1-Score* (*k* = 26)**

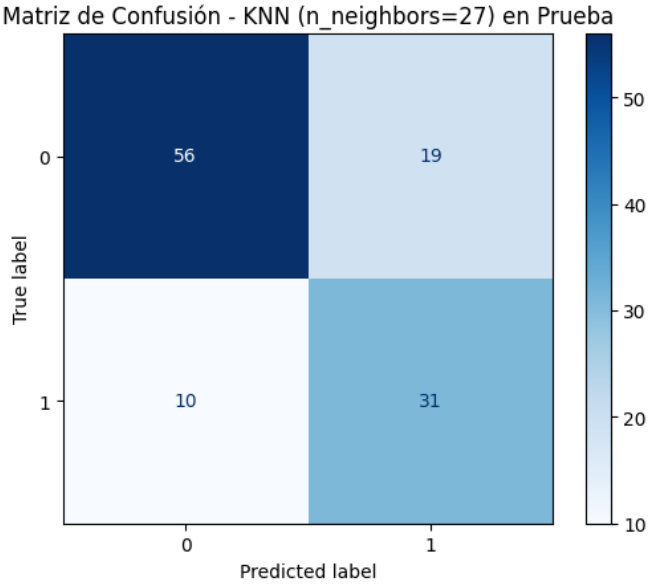

Fig. 12. Matriz de Confusión Recall

- **Accuracy**: 0.7586
- **Precision**: 0.6327
- **Recall**: 0.7561
- **F1-Score**: 0.6889

El modelo basado en *F1-Score* balancea tanto la precisión como el *recall*, siendo ideal en situaciones donde ambos valores son igualmente importantes. La matriz de confusión fue la misma que en el modelo optimizado por *Accuracy*, mostrando 18 falsos positivos y 10 falsos negativos.

C. Anemia Dataset

1) Carga del conjunto de datos: Los datos fueron preprocesados, se contó con un total de 1421 datos y 6 columnas que serán tomadas como características para el análisis del dataset. La primera característica es **Gender** indica el género de cada paciente con un valor numérico, donde 1 representa masculino y 2 femenino; **Hemoglobin (Hgb)** es una proteína en los glóbulos rojos que transporta oxígeno a los órganos y tejidos del cuerpo, también transporta dióxido de carbono desde los órganos y tejidos de regreso a los pulmones, la cual ndica la cantidad específica de esta presente en la sangre, medida en gramos por decilitro (g/dL). Por otro lado, **Mean Corpuscular Hemoglobin (MCH)** indica la cantidad promedio de hemoglobina en un solo glóbulo rojo, medida en picogramas (pg); **Mean Corpuscular Hemoglobin Concentration (MCHC)** es la cantidad de concentración de

hemoglobina en un glóbulo rojo promedio, medida en gramos por decilitro (g/dL). También, **Mean Corpuscular Volume (MCV)** es el tamaño promedio de los glóbulos rojos, medida en femtolitros (fL) y **Result** indica si un paciente padece o no anemia con un valor numérico, donde 1 representa positivo y 0 indica negativo.

2) Exploración inicial del conjunto de datos: Exploramos la información general que ofrece el dataset junto con la descripción estadística, sin las columnas de Gender y Result, para así enfocarnos en cada característica presente. Asimismo, para tener un mejor noción de los pacientes calculamos el número de mujeres y hombres totales, cuyo resultado fue: 681 hombres representados con el número 0 y 740 mujeres representadas con el número 1.

TABLE I
ESTADÍSTICA GENERAL DE VARIABLES DE INTÉRES DEL DATASET ANEMIA

Operación	Hemoglobin	MCH	MCHC	MCV
mean	13.41	22.91	30.25	85.52
std	1.97	3.97	1.40	9.64
min	6.60	16.00	27.80	69.40
max	16.90	30.00	32.50	101.60

En la Tabla I se muestra una estadística descriptiva de las variables importantes del estudio. Podemos observar que los niveles de hemoglobina varían entre 6.60 y 16.90 g/dL, mientras que el MCH se encuentra entre 16.00 y 30.00 pg. La concentración de hemoglobina corpuscular media (MCHC) presenta un rango de 27.80 a 32.50 g/dL, y el volumen corpuscular medio (MCV) oscila entre 69.40 y 101.60 fL.

Seguidamente, se revisó la presencia de valores nulos en el conjunto de datos. Los resultados mostraron que no hay valores nulos en ninguna de las columnas. Esto es un aspecto positivo ya que significa que el conjunto de datos está completo y no es necesario realizar ninguna imputación o manejo especial para datos faltantes.

Se analizó la cantidad de valores únicos en cada columna para entender la variabilidad de los datos. La columna ”Gender” contiene dos valores únicos, indicando que es una variable categórica binaria. En cuanto a las características numéricas, la columna ”Hemoglobin” tiene 81 valores únicos, la columna ”MCH” presenta 136 valores únicos, la columna ”MCHC” tiene 48 valores únicos, y finalmente, la columna ”MCV” muestra 262 valores únicos. Esto evidencia una gran diversidad en en la mayoría de las columnas; por otro lado, la columna ”Result” también tiene dos valores únicos, similar a ”Gender”, lo que sugiere que es una característica categórica binaria.

3) Exploración detallada de características: Se hizo una visualización de las distribuciones individuales de las características con el fin de identificar patrones o anomalías en los datos. Los histogramas fueron de gran ayuda, permitiendo observar la distribución de cada variable en relación con la clase objetivo. Por ejemplo:

- Hemoglobin: presenta una distribución una variabilidad considerable, como se observa en la Figura 13, el pico más pronunciado se encuentra en el rango de 12.5 a 15.0 g/dL, mientras que hay muy pocos datos con niveles de hemoglobina inferiores a 10.0 g/dL.

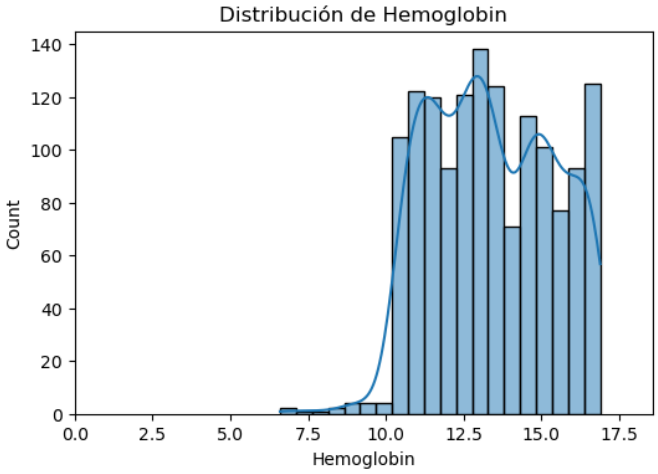

Fig. 13. Distribución de Hemoglobina

- MCV: presenta una distribución más alta que la anterior; sin embargo, también tiene una variabilidad notoria con el pico más alto con un volumen corpuscular medio aproximadamente de 100 fL.

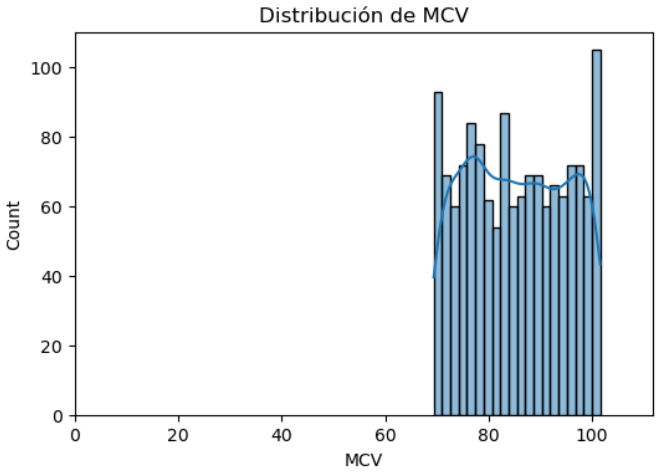

Fig. 14. Distribución de Volumen Corpuscular Medio

Además, se llevó a cabo un análisis de correlación para entender mejor las relaciones entre las características. Aunque no se observaron correlaciones extremadamente altas entre las variables, la revisión permitió una mejor comprensión del comportamiento de los datos, contribuyendo a una selección adecuada de los modelos para el análisis posterior. Véase Figura 15.

Finalmente, en la Figura 16 se presentan las relaciones entre múltiples dónde se observó que la relación entre Hemoglobina y MCV confirma que los pacientes con anemia presentan niveles bajos de hemoglobina y una variación en el volumen corpuscular medio, lo que podría indicar tipos específicos de

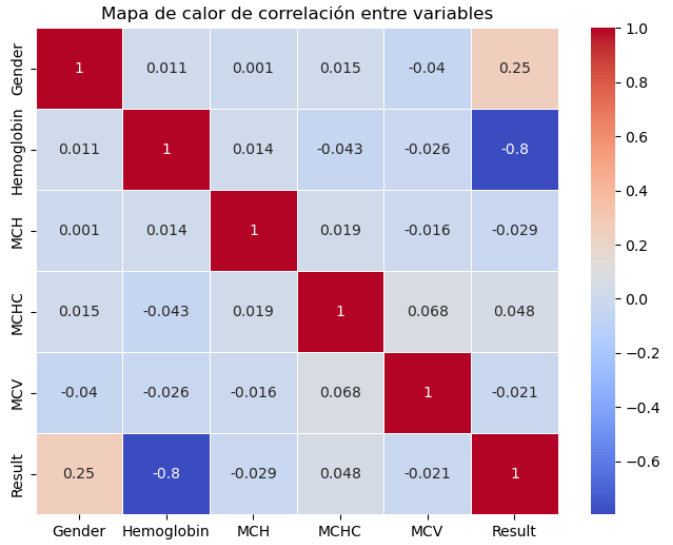

Fig. 15. Mapa de calor de correlación entre variables de Anemia

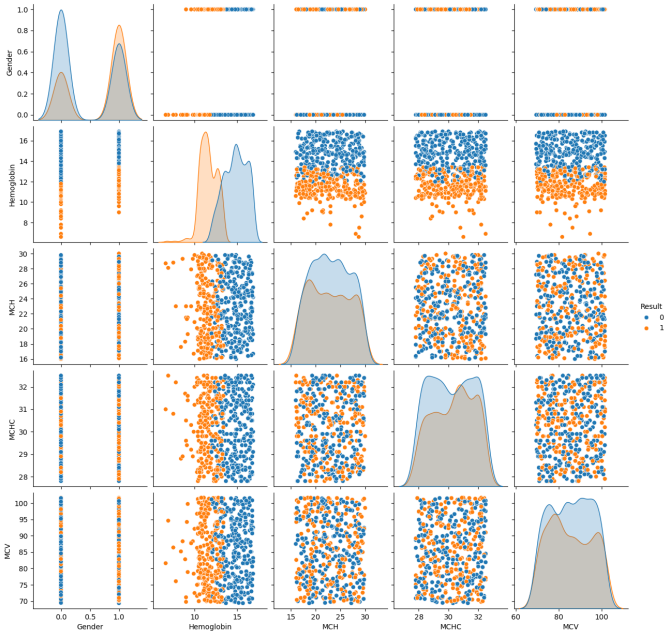

Fig. 16. Relaciones entre características de Anemia

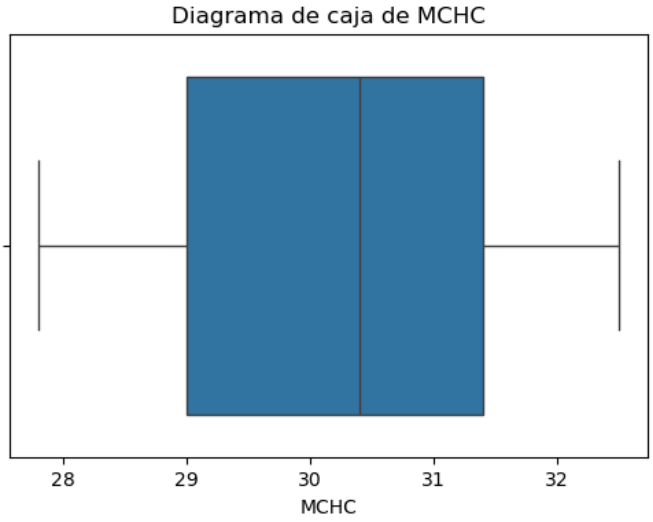

Fig. 17. Diagrama de caja de Concentración de Hemoglobina Corpuscular Media

anemia. Un recuento bajo de hemoglobina puede estar asociado con una enfermedad o afección que hace que el cuerpo tenga muy pocos glóbulos rojos [9].

Tal como se puede ver en la relación Hemoglobin y Gender, hay más mujeres que hombres con una cantidad de hemoglobina baja, esto debido a que se calcula que la anemia afecta a un 20% de los niños de 6 a 59 meses de edad, un 37% de las embarazadas y un 30% de las mujeres de 15 a 49 años [10]. Sin embargo la edad no es una característica en este dataset.

Por último, los niveles bajos de MCH en las relaciones con otros se justifica por niveles bajos de hierro en sangre (anemia), donde los niveles bajos de hemoglobina también pueden ser provocados por la talasemia (irregularidad hereditaria de la sangre) o por una enfermedad denominada Saturnismo [11].

4) Detección de outliers: Para la detección de valores atípicos, se seleccionaron las características más relevantes: Gender, Hemoglobin, MCH, MCHC y MCV. Se generaron visualizaciones para identificar posibles outliers en cada una de estas características utilizando el rango intercuartílico (IQR). En general, la mayoría de las características presentaron valores atípicos, con la excepción de Hemoglobin, que solo mostró un valor atípico, representando el 0.07% del total del conjunto de datos, como se observa en la Figura 18. Por otro lado, en la Figura 17 se ejemplifica la característica MCHC, nótese que no tiene anomalías fuera del rango intercuartílico, indicando que los datos se ajustan adecuadamente dentro de este rango con una mediana que supera los 30

5) Evaluación del balance de clases: Para analizar la distribución de las clases nos respaldamos del gráfico de barrar que muestra la frecuencia de cada clase. En la Figura 19, se puede observar cómo se distribuyen las instancias de las dos clases presentes en el conjunto de datos: ”0” para los individuos sin anemia y ”1” para aquellos con anemia.

Además del gráfico, se calcularon los porcentajes correspondientes a cada clase para obtener una visión cuantitativa de la distribución. Los resultados indican que el 56.369% de las muestras pertenecen a la clase ”0” (sin anemia), mientras que el 43.631% corresponden a la clase ”1” (con anemia). Estos porcentajes evidencian un equilibrio relativamente adecuado entre las dos clases, aunque existe una ligera desproporción a favor de la clase ”0”, por esta razón optamos por implementar una técnica de balanceo de clases para que no afecte el rendimiento de los modelos a entrenar.

6) División del conjunto de datos: Las dimensiones de los conjuntos de datos utilizados para el análisis son las siguientes: el conjunto de entrenamiento cuenta con 994 muestras, cada una de las cuales tiene 5 características. Por otro lado, el

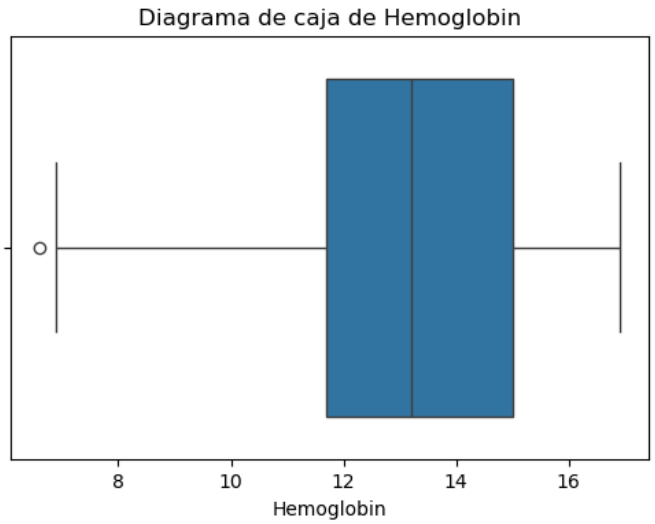

Fig. 18. Diagrama de caja de Hemoglobina

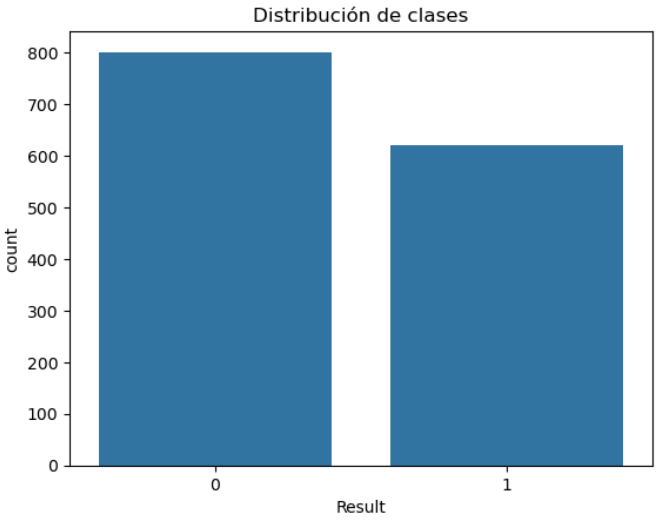

Fig. 19. Gráfico de barras para la Distribución de clases

conjunto de validación está compuesto por 213 muestras con 5 características cada una y el conjunto de prueba incluye 214 muestras, también con 5 características, las cuales se utilizan para evaluar el rendimiento final del modelo.

7) Normalización, escalabilidad y sobremuestreo: La escalabilidad de los datos se realizó teniendo en cuenta que las distribuciones de las características no siguen una distribución normal. Las suposiciones y resultados de la prueba de Shapiro-Wilk indicaron que las distribuciones no eran normales, es decir, evaluando la normalidad se obtuvo que ninguna cumplía con la premisa `p < 0.05`, lo que justifica el uso de técnicas de `MinMaxScaler`.

La aplicación de SMOTE se realizó a pesar del leve desbalance entre clases y la poco cantidad de outliers totales, con le fin de evitar que estas generen un problema en el futuro o a la hora de analizar las salidas. Como resultado, se logró un balance perfecto del 50% para ambas clases en el conjunto de datos. Esto confirma la aplicación correcta de SMOTE y su efectividad en la mejora del balance de clases. Para ilustrar el impacto de SMOTE, se presentan las características de MCH antes y después de la aplicación del método en las figuras 20 y 21. En estas figuras, se destaca el notable aumento en el tamaño del eje y, que refleja el incremento en la densidad de los datos debido al sobremuestreo.

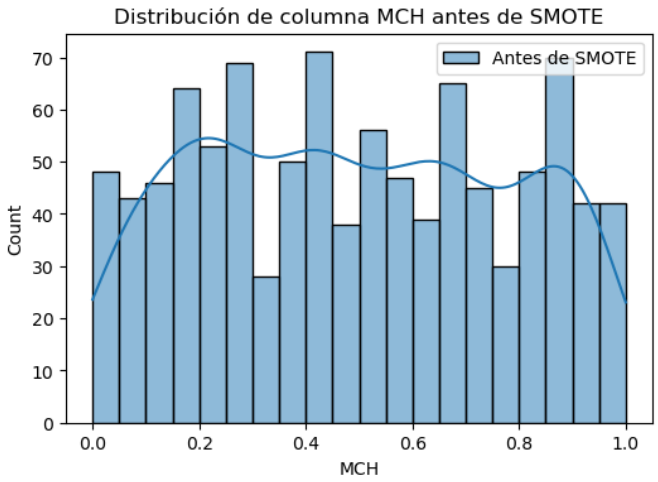

Fig. 20. Distribución Hemoglobina Corpuscular Media antes de SMOTE

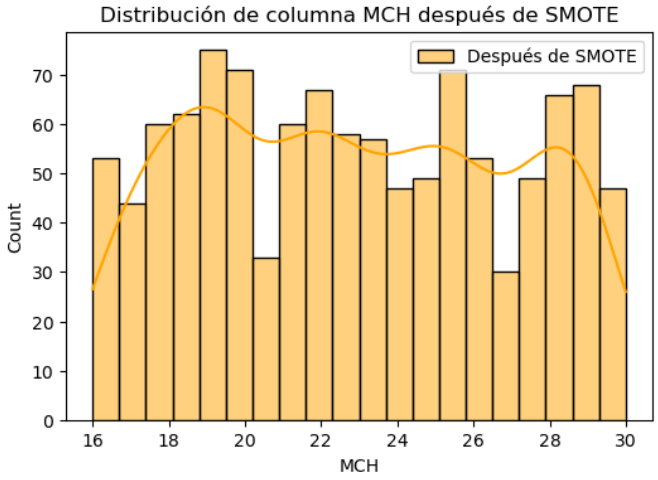

Fig. 21. Distribución Hemoglobina Corpuscular Media después de SMOTE

8) Implementación de los modelos de clasificación:

a) Regresión logística: Como primer experimento, se modeló el algoritmo con los datos obtenidos del sobremuestreo en el conjunto de datos de validación para después modelarlo con los datos escalados.

Resultados del modelo de Regresión Logística con datos sobremuestreados (Validación):

- **Accuracy:** 0.9765
- **Precision:** 0.9490
- **Recall:** 1.0000

- **F1-Score:** 0.9738
- **AUC-ROC:** 0.9998

Resultados del modelo de Regresión Logística con datos escalados (Validación):

- **Accuracy:** 0.9671
- **Precision:** 0.9300
- **Recall:** 1.0000
- **F1-Score:** 0.9637
- **AUC-ROC:** 1.0000

Ambos modelos mostraron un rendimiento sólido, con diferencias mínimas. El modelo entrenado con datos sobremuestreados presentó ligeramente mejores métricas de precisión y F1-Score, lo que sugiere que el sobremuestreo ha permitido mejorar el rendimiento en la clase minoritaria (anemia), mientras que el modelo escalado sigue siendo altamente competitivo y preciso en sus predicciones. Esto indica que tanto el manejo del desbalance como la escalabilidad de los datos mejoran el rendimiento del modelo. Por esta razón, elegimos continuar el resto de las implementaciones con los datos sobremuestreados.

En el segundo experimento, se seleccionaron ciertas combinaciones de hiperparámetros alternativos para otro entrenamiento del modelo de regresión logística, cuyos resultados fueron bastante altos. Se obtuvieron las métricas de la verificación con el conjunto de datos de validación y las métricas para evaluar el modelo en el conjunto de datos de prueba con cada combinación.

Tuvimos que la mejor combinación de hiperparámetros basada en el AUC-ROC en el conjunto de datos de validación tuvo como hiperparámetros el solver liblinear, C igual a 1 y una penalización de l1 con las siguientes métricas:

- **Accuracy:** 0.985915
- **Precision:** 0.96875
- **Recall:** 1.0
- **F1-Score:** 0.984127
- **AUC-ROC:** 1.0

Muestra un rendimiento sobresaliente. El accuracy indica que casi todas las predicciones fueron correctas, la precisión indica que fue capaz de clasificar correctamente el 96.88% de las instancias que predijo como positivas, se obtuvo un recall perfecto lo que significa que el modelo no dejó escapar ninguna instancia positiva. El F1-Score de 0.98 refleja un buen equilibrio entre precisión y recall, y el AUC-ROC de 1.0 indica una separación perfecta entre las clases.

Ahora, para el conjunto de datos de prueba, la mejor combinación basada en AUC-ROC tuvo como hiperparámetros el solver newton-cg, C igual a 1 y una penalización de l2 con las siguientes métricas:

- **Accuracy:** 0.962617
- **Precision:** 0.920792
- **Recall:** 1.0
- **F1-Score:** 0.958763
- **AUC-ROC:** 0.999733

Se observó que el accuracy disminuye en esta ocasión, lo que implica una ligera reducción en la capacidad del modelo para hacer predicciones correctas en datos no vistos previamente; la precisión también bajó, con una diferencia de aproximadamente 0.05 lo cual indica que el modelo está cometiendo más errores; un recall perfecto nuevamente con un F1-Score que representa el equilibrio con una diferencia de 0.02 disminuyendo y por último, AUC-ROC disminuyó 0.01 lo cual es muy cercano a 1, pero ligeramente inferior que en el primer experimento.

Definitivamente las diferencias entre los solvers y penalizaciones en cada conjunto pueden estar influenciadas por las características propias de los datos y la manera en que cada solver maneja las optimizaciones. Se logró representar con una matriz de confusión ambos experimentos, en la Figura 22 la matriz de confusión muestra un rendimiento perfecto del modelo, clasificando correctamente 113 casos negativos y 93 casos positivos, sin cometer errores. No hubo falsos positivos ni falsos negativos, lo que significa que todas las predicciones fueron acertadas. Por otro lado, en la Figura 23 se muestra también un rendimiento excelente. Clasificó correctamente 117 casos negativos y 93 casos positivos. Hubo solo 3 falsos positivos, pero no se detectaron falsos negativos. Esto refleja una alta precisión en las predicciones del modelo.

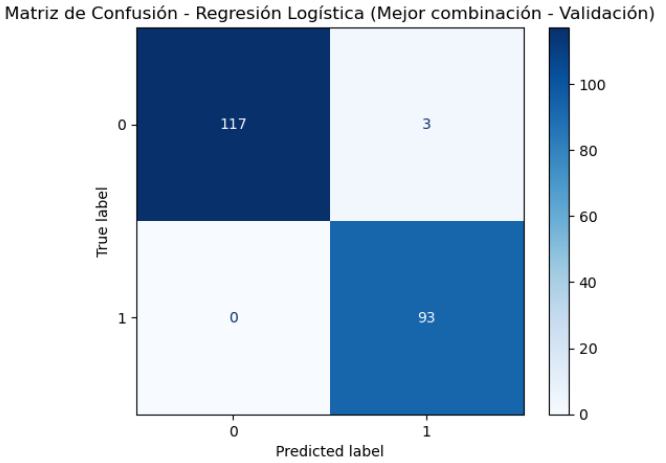

Fig. 22. Matriz de Confusión para el primer experimento

b) K-Nearest Neighbors (KNN): Durante el entrenamiento del modelo K-Nearest Neighbors (KNN) con un número de vecinos que varía entre 1 y 100, se evaluaron las métricas clave de rendimiento (accuracy, precision, recall, F1 y AUC-ROC) en el conjunto de datos de validación. A continuación, se describen los resultados obtenidos para algunos valores de nn:

Para n=1n=1, el modelo logró un alto rendimiento con un accuracy de 0.9671, una precision de 0.9300 y un recall perfecto de 1.0000, lo que resultó en un F1 de 0.9637 y un AUC-ROC de 0.9708. Sin embargo, este rendimiento extremadamente alto con un solo vecino sugiere un posible sobreajuste, ya que el modelo puede estar memorizando los datos en lugar de generalizar bien para nuevos ejemplos. Conforme aumentó el valor de nn, se observaron algunas disminuciones en estas métricas, lo que puede indicar una mejor generalización. Por ejemplo, con n=5n=5, el accuracy bajó a 0.8498, la precision

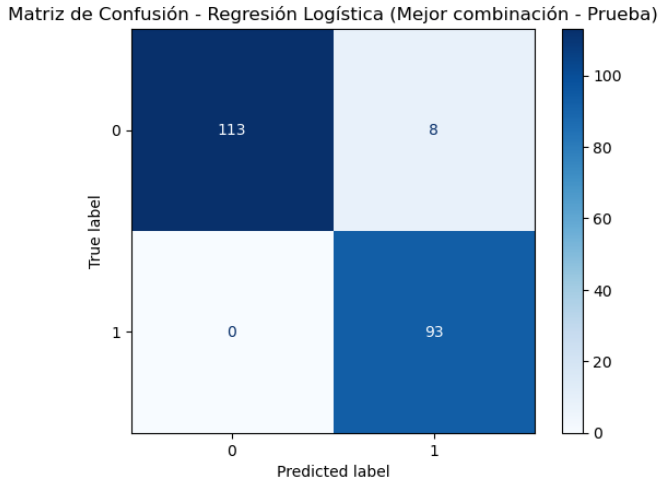

Fig. 23. Matriz de Confusión para el segundo experimento

fue de 0.8081 y el F1-score disminuyó a 0.8333.

A medida que el número de vecinos aumentaba, el rendimiento del modelo en general se mantenía estable con pequeñas variaciones. Para valores más altos de nn, como n=100n=100, el accuracy fue de 0.8122, la precision de 0.7387 y el AUC-ROC de 0.9146. Esto muestra que el modelo comienza a perder precisión y sensibilidad, pero también es menos probable que esté sobreajustado, ya que hace predicciones más equilibradas.

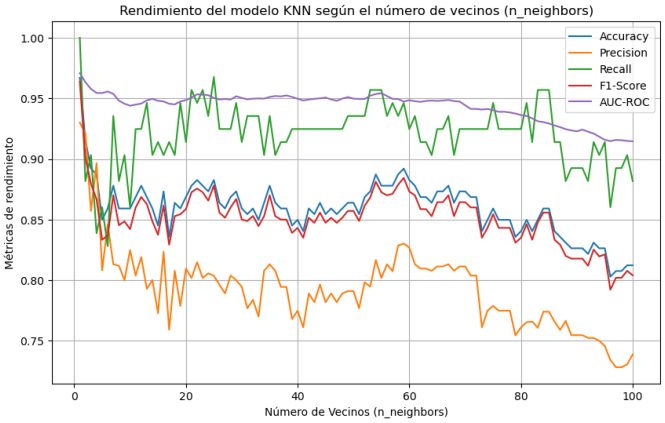

Fig. 24. Rendimiento de KNN según el número de vecinos con sus respectivas métricas

La Figura 24 muestra cómo varían las métricas de rendimiento (accuracy, precision, recall, F1-score y AUC-ROC) del modelo K-Nearest Neighbors (KNN) en función del número de vecinos. A medida que el valor de n aumenta, se observa una disminución en algunas métricas, como la precisión y el F1-score, mientras que otras, como el AUC-ROC, se mantienen relativamente estables. Estos resultados sugieren que un número bajo de vecinos puede llevar a un sobreajuste, mientras que valores más altos tienden a mejorar la generalización, aunque a costa de un menor rendimiento en las métricas específicas.

A partir de los resultados obtenidos en el análisis del número óptimo de vecinos (n_neighbors) para el modelo K-Nearest Neighbors, se seleccionaron los mejores valores de este hiperparámetro para cada métrica de rendimiento: accuracy, precisión, recall, F1-score y AUC-ROC en el conjunto de validación. Para cada una de estas métricas, se identificó el número de vecinos que maximizó su desempeño, resultando en diferentes valores óptimos para cada métrica, los resultados fueron: cuyos resultados fueron:

- **Accuracy:** 1
- **Precision:** 1
- **Recall:** 1
- **F1-Score:** 1
- **AUC-ROC:** 1

Con base en estos resultados, el mejor número de vecinos para maximizar el AUC-ROC fue 1. Se entrenó y evaluó el modelo KNN con este número de vecinos en el conjunto de prueba, obteniendo las siguientes métricas:

- **Accuracy:** 0.9720
- **Precision:** 0.9780
- **Recall:** 0.9570
- **F1-Score:** 0.9674
- **AUC-ROC:** 0.9702

Estos resultados indican que el modelo KNN con un solo vecino mostró un rendimiento sólido en el conjunto de prueba, con un AUC-ROC alto que refleja una excelente capacidad de discriminación entre las clases. Aunque genera la duda de si este rendimiento se debe a un posible sobreajuste, ya que un número tan pequeño de vecinos puede hacer que el modelo se ajuste demasiado a las particularidades del conjunto de entrenamiento. Esto sugiere que, aunque el modelo presenta métricas sobresalientes, es crucial considerar una evaluación adicional con diferentes conjuntos de datos o técnicas de validación cruzada para asegurar su capacidad de generalización y evitar un ajuste excesivo.

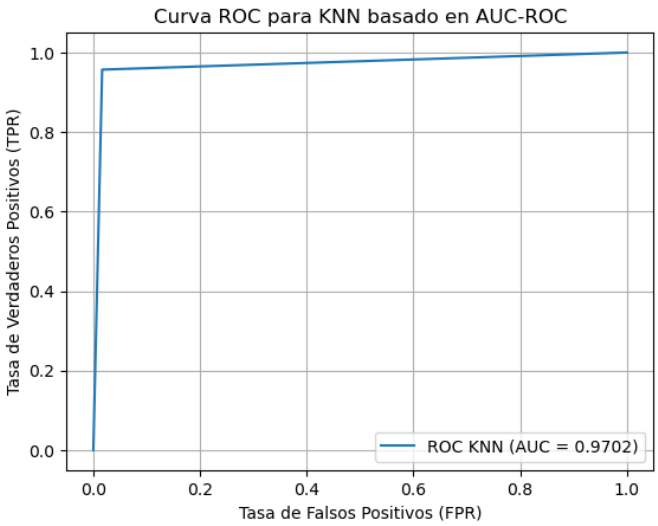

Fig. 25. Curva ROC en el modelo KNN basado en el AUC-ROC

Finalmente, en la Figura 25 presenta la curva ROC correspondiente para este modelo, ilustrando la relación entre la tasa de verdaderos positivos y la tasa de falsos positivos, y confirmando la robustez del modelo en términos de rendimiento general.

9) Interpretación y comparación de datos: Para evaluar y comparar el rendimiento de los modelos de KNN y Regresión Logística, se ha graficado la curva ROC para ambos modelos, como se muestra en la Figura 26. Ambas curvas muestran una alta capacidad de clasificación, con áreas bajo la curva (AUC) cercanas a 1, lo que sugiere que ambos modelos tienen un rendimiento sobresaliente en la tarea de predicción

Primero, se entrenó el modelo de Regresión Logística, ajustado con un máximo de 1000 iteraciones y ponderación de clases balanceada, igual que el experimento realizado con datos con sobremuestreo. Luego, se calcularon las probabilidades predichas y se graficó la curva ROC correspondiente. El valor del AUC-ROC para la Regresión Logística se obtuvo en 0.9718, indicando un excelente desempeño en la clasificación.

En comparación, el modelo KNN se entrenó utilizando el mejor número de vecinos basado en el AUC-ROC previamente determinado. La curva ROC para KNN también se graficó y el AUC-ROC para este modelo fue de 0.9702, reflejando un rendimiento muy competitivo y cercano al de la Regresión Logística.

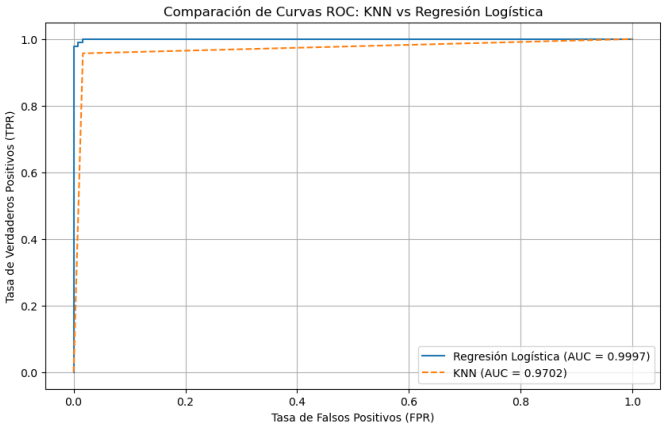

Fig. 26. Curvas ROC de los modelos KNN y Regresión Logística

IV. Discusión

En esta sección, analizaremos los resultados obtenidos en el proyecto, interpretando los datos y el desempeño de los modelos aplicados, y discutiendo sus implicaciones en el contexto de la predicción de diabetes y anemia. Además, abordaremos los posibles puntos de mejora y las limitaciones que se presentaron durante el proceso.

A. Diabetes Dataset

1) Exploración y Preprocesamiento de Datos: El preprocesamiento de los datos fue un paso crucial para garantizar que los modelos aplicados ofrecieran resultados precisos y significativos. Durante la exploración inicial de los datos, se identificaron valores anómalos en varias variables fisiológicas claves como *Glucose* (nivel de glucosa), *BloodPressure* (presión arterial), *SkinThickness* (grosor del pliegue cutáneo), *Insulin* (insulina) y *BMI* (índice de masa corporal). Los valores de 0 en estas variables fueron considerados imposibles desde un punto de vista fisiológico y tratados como valores faltantes.

Para corregir estos valores faltantes, se aplicaron técnicas de imputación. Las variables *Glucose* y *BMI*, que no presentaban tantos valores extremos, fueron imputadas utilizando la media, mientras que para variables con distribuciones más sesgadas como *BloodPressure*, *SkinThickness*, e *Insulin*, se utilizó la mediana, evitando así el sesgo introducido por los valores atípicos. Este enfoque permitió mantener la integridad del conjunto de datos sin eliminar instancias completas, que habrían reducido el tamaño de los datos disponibles.

Un aspecto crítico a considerar es que, aunque la imputación mitigó la pérdida de información, también pudo haber introducido cierta homogeneidad en las variables imputadas. Esto puede haber afectado la capacidad de los modelos para capturar patrones más complejos y relaciones no lineales entre las variables, afectando su desempeño en la fase de predicción. Este es un punto importante a tener en cuenta cuando se evalúan los resultados de los modelos, particularmente en variables como *Insulin*, que mostró una fuerte imputación de datos faltantes.

2) Análisis de Correlación y Variables Más Relevantes: El análisis de correlación permitió identificar qué variables tenían la relación más fuerte con la variable objetivo *Outcome*, que indica si un paciente tiene diabetes (1) o no (0). La variable *Glucose* mostró una correlación positiva más fuerte con la diabetes, lo que sugiere que los niveles altos de glucosa son un buen predictor para determinar si un paciente tiene diabetes, como también se ha demostrado en estudios médicos previos.

Por otro lado, *BMI* y *Age* también mostraron correlaciones positivas moderadas, lo que indica que la edad y el sobrepeso también juegan un papel importante en la predicción de diabetes, aunque no tan significativo como la glucosa. Estas variables reflejan patrones esperados, ya que la diabetes es comúnmente asociada con la obesidad y la edad avanzada.

Sin embargo, la variable *Insulin* no mostró una correlación clara con la variable objetivo. Esto podría estar relacionado con la gran cantidad de datos faltantes en esta columna y la imputación que se aplicó. En futuros estudios, obtener datos más completos sobre los niveles de insulina podría mejorar la capacidad predictiva de los modelos, especialmente dado que la insulina es un factor clínico crucial para diagnosticar y manejar la diabetes.

3) Balanceo de Clases y Uso de SMOTE: El conjunto de datos estaba desbalanceado, con aproximadamente un 65% de los registros correspondientes a pacientes sin diabetes y solo el 35% con diabetes. Este desbalance pudo haber afectado el rendimiento de los modelos, especialmente su capacidad para detectar correctamente los casos positivos (pacientes con diabetes). Los modelos tienden a favorecer la clase mayoritaria en estos escenarios, lo que puede llevar a una baja tasa de detección de la clase minoritaria.

Para mitigar este efecto, se utilizó la técnica de sobremuestreo sintético *SMOTE* (Synthetic Minority Over-sampling Technique), que generó ejemplos sintéticos de la clase minoritaria para equilibrar la distribución de las clases en el conjunto de entrenamiento. La aplicación de SMOTE mejoró las métricas de *recall* en ambos modelos, ya que permitió identificar correctamente más casos de diabetes. Este aumento en *recall* es importante en el contexto clínico, donde es preferible detectar más casos positivos, aunque implique algunos falsos positivos.

A pesar de estas mejoras, el uso de SMOTE también introdujo algunos falsos positivos adicionales, lo que afectó la precisión de los modelos. Esta compensación es común al balancear clases desiguales, y debe ser cuidadosamente ajustada para minimizar el impacto de diagnósticos incorrectos en un entorno clínico.

4) Desempeño de la Regresión Logística: La regresión logística fue uno de los modelos implementados para la clasificación de los datos. Se probaron diferentes configuraciones de hiperparámetros, incluidas la regularización L2 y la combinación ElasticNet (L1 y L2), lo que permitió controlar el sobreajuste y mejorar la selección de características relevantes. El mejor resultado se obtuvo utilizando ElasticNet con un *AUC-ROC* de 0.8728, lo que indica una capacidad excelente del modelo para discriminar entre pacientes con y sin diabetes.

En términos de *recall*, el modelo de regresión logística también mostró un desempeño consistentemente alto, lo que es un resultado positivo en un entorno clínico donde es crítico detectar la mayor cantidad posible de casos de diabetes. No obstante, el modelo también generó algunos falsos positivos, lo que se reflejó en una disminución en la *precision*. Este comportamiento podría llevar a diagnósticos innecesarios, lo que debe manejarse con cuidado en aplicaciones médicas.

Una ventaja clave de la regresión logística es su interpretabilidad. En un entorno médico, es importante que las decisiones basadas en modelos sean explicables, y la regresión logística ofrece esta ventaja, lo que facilita su adopción en entornos clínicos.

5) Desempeño de K-Nearest Neighbors (KNN): El algoritmo KNN mostró un desempeño competitivo, pero inferior en comparación con la regresión logística, especialmente en términos de precisión. Aunque KNN logró un *recall* alto, indicando que pudo identificar correctamente a los pacientes con diabetes, su precisión fue menor debido a la generación de más falsos positivos.

El valor óptimo de vecinos (k) se encontró en el rango de 26 a 28, lo que proporcionó un equilibrio razonable entre *accuracy*, *precision*, y *recall*. Sin embargo, valores más altos de k suavizaron demasiado el modelo, lo que resultó en una pérdida de detalles importantes para identificar correctamente los casos positivos de diabetes. La sensibilidad del algoritmo a la normalización de los datos también afectó su rendimiento, especialmente en la presencia de variables con valores atípicos.

En general, KNN es un modelo más adecuado para conjuntos de datos con distribuciones bien definidas y equilibradas, pero en este caso, la naturaleza desbalanceada y los valores imputados afectaron su capacidad para superar a la regresión logística.

6) Implicaciones Clínicas: En términos clínicos, la regresión logística ajustada con ElasticNet demostró ser la mejor opción para predecir la diabetes en este conjunto de datos. Su alta capacidad para discriminar entre pacientes con y sin diabetes, junto con su mejor rendimiento en métricas clave como el *AUC-ROC*, lo convierte en un modelo confiable para apoyar la toma de decisiones médicas.

Aunque KNN mostró un buen rendimiento en términos de *recall*, su menor precisión lo hace menos adecuado para escenarios donde es crucial minimizar los falsos positivos, como en la detección médica. En este contexto, la regresión logística ofrece una mejor compensación entre la sensibilidad y la precisión, lo que la hace más adecuada para aplicaciones clínicas.

7) Limitaciones y Áreas de Mejora: A pesar de los buenos resultados, el proyecto tiene algunas limitaciones. La imputación de valores faltantes, especialmente en la variable *Insulin*, pudo haber afectado la capacidad predictiva del modelo. Sería recomendable, en futuros estudios, contar con un conjunto de datos más completo y sin tantos valores faltantes, lo que permitiría una evaluación más precisa de las relaciones entre las variables.

Además, la aplicación de técnicas más avanzadas como redes neuronales profundas podría mejorar aún más el rendimiento del modelo, especialmente en la detección de casos difíciles. También sería valioso investigar enfoques adicionales para manejar los valores atípicos y mejorar la robustez de los modelos frente a la variabilidad en los datos.

B. Anemia Dataset

1) Comparación de Modelos de KNN y Regresión Logística: Ambos modelos, K-Nearest Neighbors (KNN) y Regresión Logística, demostraron un rendimiento destacado en términos de AUC-ROC y otras métricas de evaluación. Sin embargo, la evaluación detallada de cada modelo muestra ciertas diferencias que son importantes al considerar la generalización y estabilidad del rendimiento en datos no vistos.

Para el modelo KNN, el AUC-ROC fue de 0.9702, mientras que la Regresión Logística, en su mejor configuración, alcanzó un AUC-ROC de 1.0 durante la validación. Esto sugiere que la Regresión Logística fue capaz de discriminar perfectamente entre las clases en el conjunto de validación, algo que no fue tan pronunciado en el modelo KNN. A pesar de que ambos modelos muestran una alta capacidad discriminativa, el KNN podría ser más propenso al sobreajuste, debido a su dependencia directa de los datos de entrenamiento, lo cual fue evidente con métricas perfectas en algunos experimentos, lo que indica que es necesario ajustarlo cuidadosamente.

El modelo de Regresión Logística, por otro lado, mostró una mayor consistencia. En particular, con datos sobremuestreados, el modelo logró un accuracy de 0.9765, una precisión de 0.9490, y un F1-Score de 0.9738. Estas métricas sugieren que el modelo puede manejar mejor el desequilibrio de clases y mantener una alta capacidad de generalización. No obstante,

al utilizar los datos escalados, aunque las métricas bajaron ligeramente, el rendimiento general se mantuvo robusto con un AUC-ROC de 1.0. Esto refleja que la Regresión Logística es menos susceptible al sobreajuste que KNN, particularmente en escenarios donde la escalabilidad y la precisión son cruciales.

En los experimentos posteriores con combinaciones de hiperparámetros, la Regresión Logística mantuvo un rendimiento sólido. En el conjunto de validación, el solver liblinear con penalización L1 mostró una capacidad sobresaliente, con un AUC-ROC de 1.0, un recall perfecto, y un F1-Score de 0.98. Pero, en el conjunto de prueba, el solver newton-cg con penalización L2 produjo métricas ligeramente inferiores, como un accuracy de 0.9626 y una precisión de 0.9207. Esto indica que, aunque la Regresión Logística ofrece una excelente capacidad de generalización, puede haber ligeras diferencias dependiendo de los solvers y la penalización utilizada. Estas diferencias también pueden reflejar las características particulares de los datos y cómo cada solver maneja la optimización.

2) Consideraciones sobre el sobreajuste: Es importante destacar que, en ambos modelos, el sobreajuste sigue siendo una preocupación latente. Aunque el KNN presentó indicadores más obvios de sobreajuste en el conjunto de validación debido al valor del número de vecinos, la Regresión Logística, en sus mejores configuraciones, podría estar sobreajustando ligeramente en ciertas situaciones. Un aspecto clave que podría estar contribuyendo al sobreajuste es la falta de algunas características adicionales en los datos, como la edad de los pacientes, que podrían proporcionar información relevante y mejorar la capacidad predictiva del modelo. Incorporar más variables puede no solo mejorar la precisión, sino también reducir el riesgo de que los modelos aprendan patrones demasiado específicos de los datos de entrenamiento, favoreciendo una mayor generalización.

3) Implicaciones prácticas: En el contexto de predecir si un paciente tiene anemia o no, la Regresión Logística demuestra ser una opción sólida. Su rendimiento consistente y su bajo riesgo de sobreajuste la convierten en una herramienta fiable para clasificar los casos de anemia, proporcionando una buena estabilidad en la generalización de los resultados a datos no vistos. La alta precisión y el AUC-ROC cercano a 1 indican que puede distinguir eficazmente entre pacientes anémicos y no anémicos.

Por otro lado, KNN ofrece una buena capacidad discriminativa, evidenciada por su AUC-ROC superior. Sin embargo, es más propenso al sobreajuste, especialmente si el número de vecinos no está bien ajustado o es relativamente bajo. Aunque puede adaptarse a patrones más complejos en los datos, es crucial aplicar una validación rigurosa para evitar que el modelo aprenda excesivamente de las peculiaridades del conjunto de datos de entrenamiento.

En la práctica, Regresión Logística puede ser preferida para problemas de clasificación binaria como la detección de anemia debido a su estabilidad y menos riesgo de sobreajuste. KNN puede ser útil si se cuenta con datos que presentan patrones complejos y se realiza un ajuste adecuado, pero siempre con una validación exhaustiva para garantizar que el modelo no se sobreajuste a los datos específicos de entrenamiento. Además, la inclusión de características adicionales, como la edad, podría mejorar aún más la capacidad predictiva de ambos modelos.

V. Conclusiones

A. Diabetes Dataset

El presente estudio tuvo como objetivo comparar el desempeño de dos modelos de clasificación, regresión logística y K-Nearest Neighbors (KNN), para la predicción de diabetes en un conjunto de datos proporcionado por el Instituto Nacional de Diabetes y Enfermedades Digestivas y Renales. A lo largo de este proyecto, se realizaron diversos procesos de preprocesamiento de datos, ajuste de hiperparámetros y evaluación de modelos con el fin de identificar el algoritmo más adecuado para esta tarea, basada en métricas como precisión, *recall*, *F1-score* y *AUC-ROC*.

En términos generales, los resultados confirman la hipótesis planteada al inicio del artículo: **el modelo de regresión logística ofreció un mejor rendimiento global en comparación con KNN**. Esto se observó principalmente en métricas como el *AUC-ROC*, donde la regresión logística ajustada con ElasticNet alcanzó un valor de 0.8728, superando a KNN en su capacidad de discriminar entre pacientes con y sin diabetes. La regresión logística también destacó en términos de precisión y *recall*, lo que la hace más apropiada para un entorno clínico, donde es crucial minimizar tanto los falsos negativos como los falsos positivos.

El modelo de KNN, por su parte, mostró un desempeño competitivo en términos de *recall*, identificando correctamente a un número considerable de pacientes con diabetes. Sin embargo, su precisión fue menor debido a la generación de más falsos positivos. Esto se debe en parte a la sensibilidad de KNN a la escala de las variables y al desbalance de clases, características que afectan su rendimiento en comparación con la regresión logística.

A lo largo del análisis, se identificó que las variables más influyentes en la predicción de la diabetes fueron los niveles de glucosa y el índice de masa corporal (*BMI*). Esto concuerda con estudios clínicos previos, que han identificado la glucosa como un indicador clave para el diagnóstico de la enfermedad. Sin embargo, la variable insulina, a pesar de ser clínicamente relevante, no mostró ser un buen predictor en este análisis, lo que podría estar relacionado con la gran cantidad de valores faltantes que fueron imputados.

El uso de la técnica de sobremuestreo *SMOTE* para balancear las clases también fue crucial para mejorar el rendimiento de los modelos, especialmente en términos de *recall*. Sin embargo, esta técnica introdujo algunos falsos positivos adicionales, afectando la precisión de ambos modelos. Esto refleja una de las principales limitaciones de este proyecto: la compensación entre precisión y *recall* al trabajar con conjuntos de datos desbalanceados.

En cuanto a la hipótesis planteada sobre el valor óptimo de k en el modelo de KNN, se esperaba que el mejor valor de k estuviera en el rango de 5 a 7, lo cual no se corroboró con los resultados. El análisis mostró que los valores óptimos de k

se encontraban entre 26 y 28, dependiendo de la métrica de rendimiento utilizada (*accuracy*, *precision*, *recall* o *F1-score*). Estos valores superiores a los predichos sugieren que el modelo necesitó considerar un número mayor de vecinos para lograr un mejor equilibrio en la clasificación, probablemente debido a la naturaleza compleja y dispersa de los datos después del preprocesamiento y el uso de SMOTE.

En conclusión, **la regresión logística ajustada con Elastic-Net demostró ser el mejor modelo para predecir la diabetes en este conjunto de datos**, ofreciendo un equilibrio adecuado entre precisión y *recall*, lo que la hace más confiable para un entorno clínico. KNN, aunque mostró un buen desempeño en ciertas métricas, es menos adecuado para problemas donde las variables predictoras tienen escalas y distribuciones variadas.

B. Anemia Dataset

Ambos modelos, Regresión Logística y KNN, demostraron un rendimiento sólido en la tarea de predecir la presencia de anemia. Ambos arrojaron valores elevados en métricas clave como el AUC-ROC, lo que sugiere que fueron capaces de discriminar correctamente entre pacientes con y sin anemia. La Regresión Logística se destacó por su estabilidad, manteniendo un riesgo menor de sobreajuste al trabajar con datos sobremuestreados. El KNN, por su parte, mostró una ligera ventaja en su capacidad de discriminación de clases, aunque presentó indicios de sobreajuste, lo que sugiere que es más susceptible a aprender patrones específicos del conjunto de datos de entrenamiento.

El uso del sobremuestreo resultó ser una estrategia efectiva, particularmente para la Regresión Logística, que mejoró significativamente en términos de precisión y F1-score. Esto indica que el sobremuestreo ayudó a abordar el desbalance de clases, permitiendo una mayor precisión en la clasificación de casos de anemia sin comprometer la generalización. En el caso de KNN, aunque el sobremuestreo también lo benefició, los resultados sugieren que es necesario un ajuste más cuidadoso para evitar que el modelo aprenda en exceso las características del conjunto de entrenamiento.

Las características seleccionadas fueron claves para los resultados obtenidos. Sin embargo, es posible que falten características adicionales que podrían mejorar la capacidad predictiva de ambos modelos. La inclusión de variables como la edad de los pacientes, por ejemplo, podría aportar un valor significativo, dado que la edad está estrechamente relacionada con la anemia y otras condiciones de salud. Estas características adicionales podrían contribuir a mejorar la precisión y la capacidad de generalización de los modelos.

En la práctica, la Regresión Logística se presenta como una opción preferible cuando se busca estabilidad y facilidad de interpretación en un problema de clasificación binaria como la detección de anemia. Su bajo riesgo de sobreajuste la convierte en una herramienta confiable para aplicaciones clínicas, donde es fundamental contar con un modelo preciso y comprensible. El KNN, en cambio, podría ser útil en escenarios donde se requiera una mayor capacidad de discriminación, siempre y cuando se ajuste cuidadosamente para evitar el sobreajuste. Con el ajuste adecuado de sus hiperparámetros, el KNN puede ofrecer ventajas en situaciones con relaciones más complejas entre las variables y las clases.

REFERENCES

[1] IBM, "¿qué es la inteligencia artificial en la medicina?" 2023. [Online]. Available: https://www.ibm.com/mx-es/topics/artificial-intelligence-medicine

[2] Organización Panamericana de la Salud, "Diabetes," 2021. [Online]. Available: https://www.paho.org/es/temas/diabetes

[3] National Heart, Lung and Bood Institute, "¿qué es la anemia?" 2021. [Online]. Available: https://www.nhlbi.nih.gov/es/salud/anemia

[4] Instituto Nacional de Diabetes y Enfermedades Digestivas y Renales, "Pima indians diabetes database," 2016. [Online]. Available: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

[5] B. Ranjan, "Anemia dataset," 2022. [Online]. Available: https://www.kaggle.com/datasets/biswaranjanrao/anemia-dataset

[6] Amazon Web Services, "¿qué es la regresión logística?" 2022. [Online]. Available: https://aws.amazon.com/es/what-is/logistic-regression/

[7] IBM, "¿qué es knn?" 2023. [Online]. Available: https://www.ibm.com/mx-es/topics/knn

[8] MIOTI, "¿qué son los outliers?" 2023. [Online]. Available: https://mioti.es/es/que-son-los-outliers/

[9] Clínica Mayo, "Recuento de hemoglobina bajo," 2022. [Online]. Available: https://www.mayoclinic.org/es/symptoms/low-hemoglobin/basics/causes/sym-20050760

[10] Organización Global de la Salud, "Anemia," 2020. [Online]. Available: https://www.who.int/es/health-topics/anaemia

[11] VIAMED, "¿qué es el hcm?" 2024. [Online]. Available: https://www.viamedsalud.com/pruebas-medicas/analisis-clinicos/que-es-hcm-analisis-de-sangre/

Conjunto de datos de Diabetes		
Criterios	Puntuación máxima	Puntuación obtenida
Análisis del conjunto de datos y features	5	
Análisis de regresión logística	15	
Análisis de KNN	15	
Comparación de modelos	10	
Conjunto de datos seleccionado		
Criterios	Puntuación máxima	Puntuación obtenida
Análisis del conjunto de datos y features	5	
Análisis de regresión logística	15	
Análisis de KNN	15	
Comparación de modelos	10	
Aspectos Generales		
Criterios	Puntuación máxima	Puntuación obtenida
Completitud de entregables	5	
Estructura de artículo científico	5	
Aspectos Extra (SRE)		
Criterio	Puntuación máxima	Puntuación obtenida
Usar Gitflow como proceso de colaboración y utilizar tags de versionamiento (main branch). Deben participar los 2 integrantes.	5	
Total	105	