

# Modelo transformer para la identificación de la nefropatía diabética en pacientes Mexicanos

Luis Ramón Tercero Martínez González, José Adán Hernández Nolasco

Universidad Juárez Autónoma de Tabasco,  
División Académica de Ciencias y Tecnologías de la Información,  
México

232h21004@alumno.ujat.mx, adan.hernandez@ujat.mx

**Resumen** Recientes avances en inteligencia artificial han permitido el desarrollo de modelos de aprendizaje profundo para la identificación de enfermedades. La nefropatía diabética es una de las principales complicaciones de la diabetes mellitus tipo 2, y su identificación es fundamental para prevenir el deterioro renal. Este estudio propone un modelo transformer como apoyo del diagnóstico de la nefropatía diabética en pacientes mexicanos. Es un modelo que utiliza el mecanismo de autoatención, el cual permite capturar relaciones complejas entre variables clínicas. Se aplicaron técnicas de preprocesamiento para garantizar la calidad de los datos y se utilizó una técnica de incrustación para representar similitudes entre variables. Así mismo, fue posible obtener una interpretación del resultado final del modelo, por medio de la extracción de los pesos de atención, obtenidos de los cálculos matemáticos internos del modelo. Además, se logró en la evaluación una precisión de 0.99% en la clasificación de la enfermedad. Los hallazgos sugieren que este enfoque moderno puede ser una herramienta de apoyo efectiva y confiable para el diagnóstico de la nefropatía diabética en la población mexicana.

**Palabras clave:** autoatención, incrustaciones, datos tabulares, transformadores.

## Transformer Model for the Identification of Diabetic Nephropathy in Mexican Patients

**Abstract.** Recent advances in artificial intelligence have enabled the development of deep learning models for disease detection. Diabetic nephropathy is one of the main complications of type 2 diabetes mellitus, and its early identification is essential to prevent kidney deterioration. This study proposes a transformer-based model as a clinical decision support tool for diagnosing diabetic nephropathy in Mexican patients. The model leverages the self-attention mechanism, which captures complex relationships among clinical variables. Preprocessing techniques

were applied to ensure data quality, and an embedding technique was used to represent similarities between variables. Furthermore, interpretability was achieved by extracting attention weights from the model's internal computations. The evaluation yielded a classification accuracy of 99%, suggesting that this modern approach can serve as an effective and reliable tool to assist in the diagnosis of diabetic nephropathy within the Mexican population.

**Keywords:** self-attention, embeddings, tabular data, transformers.

## 1. Introducción

La inteligencia artificial (IA) es una rama de la informática que permite a las computadoras realizar tareas que normalmente requieren inteligencia humana, se ha convertido en un campo clave para resolver problemas complejos en diversas disciplinas. Además, el aprendizaje automático (AA), es una rama de la IA que permite a las computadoras aprender y mejorar de forma autónoma. Las redes neuronales son una sub rama del AA y si estas tienen un número grande de capas, se le denomina aprendizaje profundo (AP). EL AP destaca por su capacidad para extraer patrones a partir de grandes volúmenes de información, con impacto en sectores como la salud. Por lo tanto, permite abordar desafíos como el diagnóstico de enfermedades. De manera general, la arquitectura del AP incluyen una capa de entrada para recibir datos, capas intermedias ocultas para procesarlos y una capa de salida que genera el resultado [1].

En particular, las arquitecturas de AP han demostrado un alto rendimiento en la identificación de características relevantes para la predicción de enfermedades, alcanzando en precisión y generalización a los algoritmos de AA; en específico las arquitecturas basadas en transformer [2], principalmente utilizados en procesamiento de lenguaje natural. Él autor menciona en su estudio, que la arquitectura transformer del AP ha sido aplicado con éxito para procesar datos tabulares y han mostrado un rendimiento mayor al de los algoritmos de AA como los árboles [3]. Su capacidad para modelar relaciones no lineales sin necesidad de predefinir interacciones entre variables es ideal para el diagnóstico de enfermedades.

Sin embargo, la evolución del transformer para datos tabulares ha avanzado desde redes totalmente conectadas hasta arquitecturas híbridas capaces de manejar su complejidad [4]. Algunas arquitecturas emplean atención secuencial para la selección de características por instancia, mejorando la interpretabilidad, mientras que otros combinan autoatención y atención entre muestras para capturar interacciones complejas y optimizar la escalabilidad [5]. Una estrategia eficiente en este contexto es la integración de mecanismos de atención con perceptrones multicapa (MLP), como en los modelos híbridos, cuya innovación principal radica en la técnica de crear embeddings cuya función es crear vectores numéricos a los datos de entrada, lo que permite representar los datos y capturar relaciones complejas [6].

Dentro de las múltiples aplicaciones de las arquitecturas transformer en el ámbito de la salud, el diagnóstico de nefropatía diabética —una complicación grave asociada a la diabetes mellitus tipo 2 (DM2)— representa un desafío crítico. En países como México, donde la DM2 es una de las principales causas de insuficiencia renal crónica, este problema se ve agravado por la escasez de bases de datos clínicas estructuradas y la limitada adopción de herramientas de inteligencia artificial. Ante esta necesidad, el presente estudio propone una arquitectura basada en transformers capaz de capturar representaciones latentes a partir de datos clínicos tabulares, con el objetivo de mejorar la identificación temprana de factores de riesgo y apoyar el diagnóstico oportuno de la nefropatía diabética.

### **1.1. Trabajos relacionados**

En las últimas décadas, el AP ha tenido buenos resultados en la clasificación de enfermedades crónicas, como la nefropatía diabética, entre otras. Con el fin de proporcionar una visión de las investigaciones más recientes en esta aplicación. El estudio de Zisser et al [7] propone el modelo STRAFE, que explora el uso de transformers para mejorar la predicción del deterioro en pacientes con enfermedad renal crónica (ERC), reportando que el modelo mejora la identificación de pacientes de alto riesgo. Asimismo, la investigación de Arafat et al [8] propone un modelo de aprendizaje profundo para la detección automatizada de ERC, obteniendo resultados con una precisión del 99%, superando otros métodos recientes.

Los autores Singh et al [9] presentan un modelo para la mejora del diagnóstico temprano de la ERC, este trabajo utilizó AP y la técnica de selección de características mediante Recursive Feature Elimination (RFE). Con este enfoque, se identificaron variables clave que fueron utilizadas en distintos modelos de clasificación, logrando que el modelo propuesto superara a SVM, KNN, regresión logística, Random Forest y Naïve Bayes, alcanzando una precisión del 100%.

A nivel internacional, empresas como Amazon, Google, Facebook y NVIDIA han desarrollado arquitecturas híbridas que combinan AP con mecanismo de atención, optimizando el análisis de datos tabulares. Arquitecturas como GatedTabTransformer que incorporan mecanismos de gating para mejorar la selección de características [10], mientras que SAINT introduce atención entre muestras, favoreciendo la interpretabilidad de relaciones inter-muestra [11], ARM-Net emplean neuronas exponenciales y atención para un aprendizaje más eficiente de interacciones explícitas [12], TabTransformer captura interacciones complejas entre variables mediante bloques de atención y feed-forward [13], y TabNet se distingue por su preentrenamiento autosupervisado [14].

Sin embargo, se observa que muchas de las propuestas analizadas se centran en comparar los algoritmos de clasificación, sin considerar la importancia de las características significativas resultantes, la interpretabilidad del modelo y cabe resaltar que ninguno de los mencionados utiliza datos de pacientes mexicanos. Por lo tanto, en este estudio, la arquitectura propuesta es entrenada con datos

de pacientes mexicanos, convirtiéndola en una herramienta clave para el apoyo del diagnóstico de la ND. Asimismo, se observa un crecimiento en el desarrollo de estas arquitecturas híbridas, con una tendencia a usar arquitecturas más sofisticada y eficientes desde 2019 hasta 2023. Esto sugiere que la integración de mecanismos de atención en modelos tabulares ha ido perfeccionándose con el tiempo, priorizando tanto la interpretabilidad como la eficiencia en el manejo de datos estructurados, numéricos y categóricos.

## **1.2. Objetivo de estudio**

Este estudio tiene como propósito identificar a pacientes con DM2 que presentan riesgo de desarrollar ND mediante el uso de arquitecturas basado en transformer. Se emplea un clasificador entrenado con el conjunto de datos DiabetIA [17], el cual fue obtenido a través de un proyecto de acceso abierto del Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONACYT). Los datos provienen del Sistema de Informática de Medicina Familiar (SIMF) del Instituto Mexicano del Seguro Social (IMSS) en Michoacán.

## **2. Metodología**

Siguiendo la metodología CRISP-DM, se extrajeron de la base de datos DiabetIA, los datos crudos de los pacientes con DM2 y asimismo aplicando filtros, los datos de pacientes positivos y negativos con ND, constituyendo así la muestra de estudio. La muestra utilizada incluye información sobre otras comorbilidades asociadas a la DM2, tales como retinopatía, neuropatía y pie diabético. Los datos de la muestra estaban organizados en cinco categorías: demográficos, medicamentos, laboratorios, medidas y diagnóstico médico. A través de esta estructura, el estudio se centra específicamente en la clasificación de pacientes con ND, aplicando modelos basados en Transformers.

### **2.1. Preprocesamiento de la base de datos**

La muestra inicial contenía 239 columnas y 50,936 observaciones, ya etiquetadas y anonimizadas. No obstante, se eliminaron las filas con más del 50% de datos faltantes y la columna CURP, por no aportar valor predictivo al modelo. Como resultado, se conservaron 238 columnas y 19,857 observaciones. La base de datos presentaba un desbalance significativo entre clases, además de valores nulos. Para mitigar el sesgo en el entrenamiento del modelo, se aplicó una estrategia de submuestreo: se seleccionaron las 5,000 observaciones más completas del grupo de pacientes negativos (de un total de 18,689) y se concatenaron con las 3,557 observaciones correspondientes a pacientes positivos. Este conjunto balanceado de 8,557 observaciones se utilizó como base para la fase de entrenamiento y evaluación del modelo.

Además, se realizó un análisis de datos, las distribuciones, correlaciones entre variables, valores atípicos y sesgos de las variables numéricas y categóricas para

comprenderlos y nos ayuda a seleccionar las técnicas y modelo adecuado a usar. Se identificaron valores nulos en variables y en las variables categóricas, varias columnas tenían más del 70% de valores nulos, por lo que fueron eliminadas. La base de datos al final quedo con 49 columnas (25 numéricas y 24 categóricas) y 8,557 observaciones (5,000 negativas y 3,557 positivas).

## 2.2. Imputación de datos tabulares

Las medidas estadísticas media, mediana y moda, son ampliamente utilizadas para la imputación simple de datos tabulares, pero de acuerdo a la literatura, introducen sesgo en los datos. Por lo cual se decidió utilizar la técnica de *Iterative Imputer* [18], este enfoque trabaja de forma iterativa, modelando las relaciones entre variables mediante un esquema basado en árboles de decisión. De esta manera, cada valor ausente se predice utilizando la información más relevante de los demás variables, refinando progresivamente las estimaciones en cada iteración. Además, la imputación se realizó por separado para pacientes positivos y negativos, centrándose exclusivamente en las variables numéricas, ya que las categóricas estaban completas.

## 2.3. Selección de características

Se emplearon dos técnicas: LASSO (*Least Absolute Shrinkage and Selection Operator*), efectiva en estudios clínicos [15], y RFE (*Recursive Feature Elimination*) con *RandomForestClassifier*, estas técnicas son utilizadas con éxito en diagnósticos basados en datos clínicos [16]. Se obtuvieron seis grupos de 20, 30 y 40 variables mixtas, respectivamente, para entrenar el modelo en distintos casos de estudio. LASSO tiende a seleccionar más variables categóricas, mientras que RFE favorece las numéricas, eligiendo aquellas con mayor impacto en el rendimiento. Variables como el cultivo de orina, la glucosa en sangre, la creatinina, la hemoglobina y la albúmina aparecieron de manera consistente en todos los grupos, lo que refuerza su relevancia para la clasificación de la ND. La convergencia de ambas técnicas destacan la importancia de ciertos biomarcadores en la clasificación. En la siguiente sección se evalúa el grupo de 20 por Lasso y RFE, ya que este grupo tuvo mejor desempeño y resultados en el modelo basado en Transformers.

## 3. Modelo de clasificación transformers

### 3.1. Arquitectura del modelo

La arquitectura propuesta está basada en el modelo transformer adaptado para el procesamiento de datos tabulares. Esta incorpora tres mecanismos fundamentales: el *Self-Attention*, que asigna pesos a las entradas según su relevancia contextual; la *atención multi-cabeza*, que permite capturar múltiples patrones de interacción de forma paralela; y la *Gated Linear Unit (GLU)*, que

incrementa la capacidad expresiva del modelo al combinar transformaciones lineales con una compuerta sigmoide. En conjunto, estos componentes permiten capturar relaciones complejas y filtrar información irrelevante o ruidosa en los datos.

Para una implementación eficiente, estos mecanismos se integran dentro de una clase denominada *CustomTransformerEncoderLayer*, la cual constituye el núcleo del codificador propuesto. Esta clase se repite en múltiples capas, lo que permite una representación jerárquica y profunda de las variables de entrada.

En la etapa de entrada, el modelo genera representaciones vectoriales (embeddings) tanto para variables categóricas como continuas. Las variables continuas son transformadas a una dimensión fija  $d$  mediante capas lineales, mientras que las categóricas son proyectadas a través de embeddings aprendibles. Estas representaciones son combinadas y procesadas en secuencia por el codificador transformer multicapa, seguido por una capa densa encargada de realizar la clasificación.

Con el fin de brindar mayor claridad sobre la estructura matemática del modelo y justificar su implementación, a continuación se presentan las expresiones formales que definen los principales componentes del codificador. Estas ecuaciones describen cómo se realiza el procesamiento secuencial de la información y cómo se integran los distintos mecanismos que conforman la arquitectura del modelo propuesto.

**Gated Linear Unit (GLU)** La capa *GLU* mejora la flexibilidad combinando transformaciones lineales con una compuerta sigmoide y su fórmula se muestra en la ecuación 1:

$$o = \sigma(W_z x + b_z) \odot \tanh(W_h x + b_h) + (1 - \sigma(W_z x + b_z)) \odot x, \quad (1)$$

donde:

- $x$  representa el vector de entrada.
- $W_z$  y  $W_h$  son matrices de pesos para las transformaciones lineales asociadas a la compuerta sigmoide y a la función tangente hiperbólica, respectivamente.
- $b_z$  y  $b_h$  son los vectores de sesgo correspondientes.
- $\sigma(\cdot)$  denota la función sigmoide, que actúa como compuerta para controlar el flujo de información.
- $\tanh(\cdot)$  es la función tangente hiperbólica, que introduce no linealidad.
- $\odot$  indica el producto elemento a elemento (producto Hadamard) entre vectores del mismo tamaño.
- $o$  es el vector de salida de la capa GLU, que combina información filtrada y residual para una representación más expresiva.

**Mecanismo de Self-Attention** El mecanismo de *Self-Attention* asigna pesos a las entradas según su relevancia y su fórmula se muestra en la ecuación 2:

$$\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V. \quad (2)$$

Donde:

- $Q$ ,  $K$  y  $V$  representan los tensores de consulta (\*Query\*), clave (\*Key\*) y valor (\*Value\*), respectivamente, todos derivados de la entrada  $X$  mediante proyecciones lineales.
- $QK^T$  es el producto matricial entre consultas y claves transpuestas, que mide la similitud entre elementos.
- $d_k$  es la dimensión de las claves, utilizada para escalar la similitud y estabilizar los gradientes.
- $\frac{QK^T}{\sqrt{d_k}}$  es la puntuación de atención escalada.
- $\text{softmax}(\cdot)$  convierte las puntuaciones en una distribución de probabilidad, asignando pesos de atención.
- $\text{Attention}(Q, K, V)$  es la salida del mecanismo de atención, una combinación ponderada de los valores  $V$  según los pesos calculados.

**Atención Multi-Cabeza** Para capturar múltiples patrones, la *atención multi-cabeza* aplica varias atenciones en paralelo y su fórmula se muestra en la ecuación 3:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(H_1, \dots, H_h)W^O. \quad (3)$$

Donde:

- $Q$ ,  $K$ ,  $V$  son los tensores de consulta (\*Query\*), clave (\*Key\*) y valor (\*Value\*), respectivamente, derivados de la entrada mediante proyecciones lineales.
- $H_1, \dots, H_h$  representan las salidas de las  $h$  cabezas de atención, cada una operando de manera independiente con sus propias matrices de pesos.
- $\text{Concat}(H_1, \dots, H_h)$  indica la concatenación de las salidas de todas las cabezas de atención.
- $W^O$  es la matriz de pesos utilizada para proyectar la concatenación de las cabezas de vuelta al espacio de salida original.
- $\text{MultiHead}(Q, K, V)$  es la salida final del mecanismo de atención multi-cabeza, que integra múltiples perspectivas de atención para una representación más rica y contextual.

### 3.2. Selección de hiperparámetros

La tabla 1 presenta los hiperparámetros utilizados en la evaluación final de la arquitectura. La selección de estos valores se basó en una revisión exhaustiva de literatura relacionada, en la que se identificaron configuraciones comunes y efectivas en estudios previos con tareas similares. A partir de esta base, se realizaron ajustes empíricos orientados a maximizar el rendimiento y la eficiencia computacional del modelo.

El entrenamiento de los modelos se realizó utilizando una **TPU v2-8** de Google colab gratuita pero limitada, compuesta por ocho núcleos especializados

**Tabla 1.** Configuración de hiperparámetros y su justificación.

| Hiperparámetro                          | Valor          | Justificación   |
|---|----------------|---|
| Características (col_num)               | continuas $n$  | Depende del dataset, mantiene información relevante.                    |
| Características (col_cat)               | categorías $n$ | Determina el número de embeddings.                                      |
| Dimensión del embedding (dim_embedding) | 128            | Equilibrio entre capacidad de representación y sobreajuste.             |
| Cabezas de atención (num_heads)         | 4              | Captura dependencias sin aumentar excesivamente el costo computacional. |
| Capas del codificador (num_layers)      | 4              | Profundidad suficiente sin riesgo significativo de sobreajuste.         |
| Dropout (dropout)                       | 0.2            | Previene sobreajuste manteniendo la capacidad de aprendizaje.           |
| Tamaño del batch (batch_size)           | 64             | Balance entre estabilidad del gradiente y eficiencia computacional.     |
| Épocas (num_epochs)                     | 10             | Pruebas empíricas indicaron convergencia sin sobreajuste.               |
| Tasa de aprendizaje (learning_rate)     | 0.001          | Valor estándar en Adam para convergencia estable.                       |

para operaciones tensoriales de alto rendimiento, optimizada para cargas de trabajo en TensorFlow. Esta configuración permitió una aceleración drástica del proceso experimental y en el resultado final.

Gracias a esta capacidad de cómputo especializada, el tiempo de entrenamiento se redujo de forma significativa en comparación con una GPU convencional, completando múltiples iteraciones en cuestión de segundo por época, por lo cual no es factible mencionarlas. Esta mejora sustancial en rendimiento computacional no solo favoreció una exploración más amplia del espacio de hiperparámetros, sino que también ayuda la escalabilidad y reproducibilidad del estudio.

## 4. Resultados

De los varios grupos de variables creados por los selectores Lasso y RFE, en el análisis final se utilizó solamente el grupo de 20 variables numéricas/categorías, seleccionadas por selector *Lasso* y *RFE*.

Cabe destacar que las variables seleccionadas a partir de la base de datos inicial contienen información proveniente de cinco grupos médicos principales: demográficos, medicamentos, laboratorios, medidas y diagnóstico médico. Esta diversidad de datos permitió una clasificación más precisa y robusta de los pacientes.

**Tabla 2.** Reporte de la clasificación usando 20 variables seleccionadas por Lasso.

| Class           | Precisión | Recall | F1-Score | Support |
|-----------------|-----------|--------|----------|---------|
| 0               | 0.9964    | 0.9982 | 0.9973   | 547     |
| 1               | 0.9981    | 0.9962 | 0.9971   | 521     |
| <b>Accuracy</b> |           |        | 0.9972   | 1068    |

#### 4.1. Análisis de variables obtenido por Lasso

Las primeras 11 variables corresponden a exámenes de laboratorio clave en la evaluación metabólica y renal. Las siguientes 2 variables, hemoglobina y albúmina, son biomarcadores importantes. Luego, se encuentran 2 parámetros físicos (peso y pie derecho, que pueden indicar neuropatía diabética), mientras que el grupo de medicamentos incluye una variable para el uso de antidiabéticos. Finalmente, los últimos 4 términos indican el diagnóstico de diabetes tipo 2 y sus complicaciones. En cuanto al desempeño del modelo son Test Loss = 0.0088, AUC = 99.82%, accuracy = 99.99% y specificity = 0.9982%. y además se obtuvo el reporte de la clasificación interna en la tabla 2.

La matriz de confusión:

$$\begin{bmatrix} 546 & 1 \\ 2 & 519 \end{bmatrix}$$

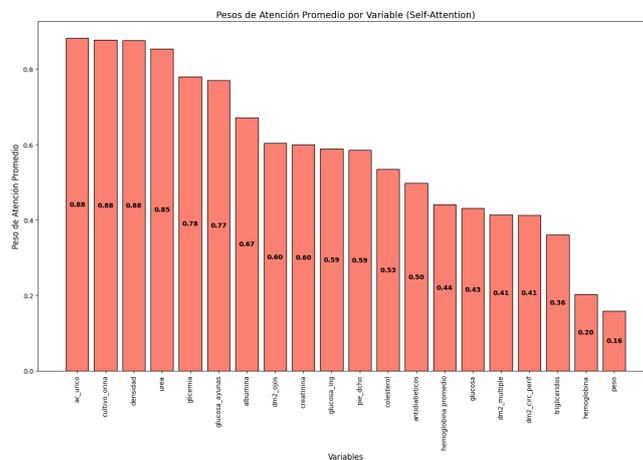
Donde:

- **546** → Verdaderos positivos (TP): Casos correctamente clasificados como sin nefropatía diabética (clase 0).
- **1** → Falsos positivos (FP): Casos incorrectamente clasificados como con nefropatía diabética (clase 1).
- **2** → Falsos negativos (FN): Casos incorrectamente clasificados como sin nefropatía diabética (clase 0).
- **519** → Verdaderos negativos (TN): Casos correctamente clasificados como con nefropatía diabética (clase 1).

Casi no se presentan falsos negativos, el modelo es 98% preciso para no pasar por alto a los pacientes con nefropatía diabética.

#### Evaluación de los pesos de atención extraídos del modelo

Se muestra en la figura 1, la interpretabilidad de las variables que el modelo prestó más atención para realizar su decisión en la clasificación. Los pesos de atención indican la importancia relativa que el modelo asigna a cada variable al realizar la clasificación.



**Fig. 1.** Grupo de 20 Variables por Lasso y su peso de atención que impactan en los resultados del modelo.

**Tabla 3.** Reporte de la clasificación usando 20 variables por RFE.

| Class           | Precisión | Recall | F1-Score | Support |
|-----------------|-----------|--------|----------|---------|
| 0               | 0.9982    | 0.9963 | 0.9973   | 547     |
| 1               | 0.9962    | 0.9981 | 0.9971   | 521     |
| <b>Accuracy</b> |           |        | 0.9972   | 1068    |

#### 4.2. Análisis de variables obtenido por RFE

Las primeras variables analizadas, incluyendo colesterol, glucosa, hemoglobina, triglicéridos, urea, creatinina, ácido úrico y cultivo de orina, son fundamentales en la evaluación de riesgos metabólicos y renales, ya que reflejan tanto el control glucémico como la función renal. La densidad urinaria y el examen general de orina (EGO) desempeñan un papel crucial en la identificación de infecciones urinarias y disfunciones renales. Por otro lado, las variables relacionadas con los pies son decisivas para detectar neuropatía diabética y complicaciones circulatorias. Asimismo, biomarcadores como la albúmina y la hemoglobina son indicadores clave de la función renal y el estado hematológico. En cuanto al desempeño del modelo son Test Loss = 0.0077, AUC = 99.99%, accuracy = 99.63%, y specificity = 0.9972% y además se obtuvo el reporte de la clasificación interna en la tabla 3.

Los resultados muestran que el modelo tiene un rendimiento excelente, con una exactitud del 99.72%. La precisión y el recall para ambas clases (0 y 1) son muy altos, lo que indica que el modelo es eficiente en la clasificación correcta de ambas clases, con muy pocos falsos positivos y falsos negativos. En particular, el F1-score de 0.9972 refleja un buen equilibrio entre precisión y recall, lo que

sugiere que el modelo no solo acierta en la mayoría de las predicciones, sino que también es eficaz para identificar tanto las instancias positivas como las negativas.

La matriz de confusión, que muestra el rendimiento de un modelo de clasificación binaria. Aquí está la interpretación:

$$\begin{bmatrix} 545 & 2 \\ 1 & 520 \end{bmatrix}$$

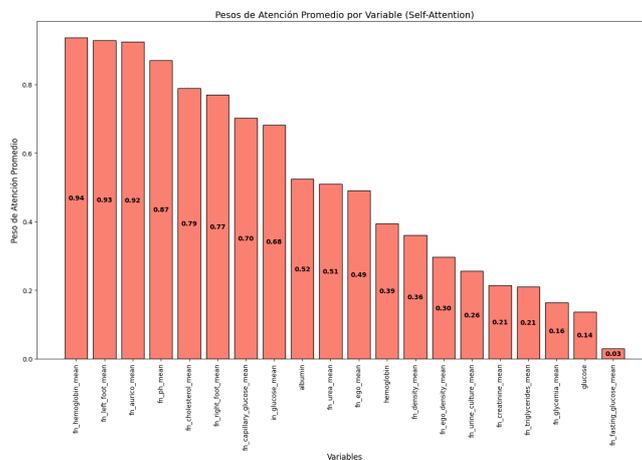
#### **Descripción de los valores:**

- **545:** Número de verdaderos positivos (TP). El modelo clasificó correctamente 545 instancias de la clase 0 (negativa).
- **2:** Número de falsos positivos (FP). El modelo clasificó incorrectamente 2 instancias de la clase 0 como clase 1.
- **1:** Número de falsos negativos (FN). El modelo clasificó incorrectamente 1 instancia de la clase 1 como clase 0.
- **520:** Número de verdaderos negativos (TN). El modelo clasificó correctamente 520 instancias de la clase 1 (positiva).

Los resultados del modelo está bien ajustado para no pasar por alto las instancias de la clase 1, que probablemente se asocian con una condición médica importante, como la nefropatía diabética. La combinación de altos verdaderos positivos y bajos falsos negativos refleja un rendimiento excepcional en la clasificación, especialmente en contextos donde la detección de la condición positiva es crítica. Esta interpretación demuestra que el modelo tiene un buen equilibrio entre sensibilidad y especificidad, mejorando la clasificación de la DM2 y minimizando errores.

#### **Evaluación de los pesos de atención extraídos del modelo**

En la figura 2 se muestran los pesos de atención reflejan la importancia que el modelo asigna a cada variable al hacer predicciones. En este caso, algunas variables presentan pesos significativamente altos, lo que indica su relevancia en las decisiones del modelo.



**Fig. 2.** Grupo de 20 Variables por RFE y su peso de atención para decidir sus resultados.

El modelo combina variables bioquímicas y clínicas, como **hemoglobina**, **urea** y **glucosa**, para tomar decisiones en la clasificación, especialmente en enfermedades renales y metabólicas. La atención elevada a **infección urinaria** resalta su relevancia en el diagnóstico.

### 4.3. Análisis de los pesos internos extraídos

Para comprender la lógica interna del modelo propuesto y validar su capacidad de interpretación clínica, se analizan los pesos asignados a cada variable durante el proceso de entrenamiento. Estos pesos reflejan la importancia relativa de cada característica en la predicción del diagnóstico de nefropatía diabética. A continuación, se presentan las variables con mayor peso, se muestran en la tabla 4, lo que permite identificar los biomarcadores más influyentes desde el punto de vista matemático y médico. Esta información es clave para establecer la concordancia entre el aprendizaje automático y el razonamiento clínico, y aporta evidencia sobre la robustez del modelo en la toma de decisiones diagnósticas.

**Validación Cruzada** Los resultados de validación cruzada utilizando 5 pliegues (KFold) muestra un rendimiento consistente del modelo, con una precisión promedio de 0.9947 y una pérdida promedio de 0.0179. Los resultados por pliegue se muestran en el tabla 5:

## 5. Conclusión y trabajo a futuro

Este estudio evaluó un modelo transformer de clasificación y se utilizó la técnica de validación cruzada para ayudar a reducir el sobreajuste, obteniendo

**Tabla 4.** Pesos internos de las variables más altas asignados por el modelo en los cálculos matemáticos.

| Variable          | Peso   | Aportación médica                    |
|-------------------|--------|--------------------------------------|
| Ácido úrico       | 0.9369 | Alta contribución a la predicción    |
| Glucosa en ayunas | 0.9369 | Alta importancia en detección        |
| Glicemia          | 0.9369 | Relevancia diagnóstica significativa |
| Hemoglobina       | 0.9369 | Indicador clave de condición general |
| Cultivo de orina  | 0.9279 | Soporte en diagnóstico secundario    |
| Urea              | 0.9245 | Alta contribución                    |
| pH (orina)        | 0.8708 | Contribución moderada                |

**Tabla 5.** Resultados de validación cruzada por pliegue.

| Pliegue | Precisión | Pérdida |
|---------|-----------|---------|
| 1       | 0.9986    | 0.0046  |
| 2       | 0.9951    | 0.0167  |
| 3       | 0.9951    | 0.0180  |
| 4       | 0.9909    | 0.0334  |
| 5       | 0.9937    | 0.0171  |

una precisión de 0.99% y una pérdida de 0.01%. Los resultados muestran su capacidad para generalizar en la clasificación de datos médicos complejos, como enfermedades metabólicas y renales.

El análisis de interpretabilidad del modelo, basado en los pesos de atención, resalta la importancia de variables metabólicas y renales en la clasificación de la nefropatía diabética. Se identificó que biomarcadores como el **ácido úrico**, **urea**, **densidad urinaria**, **glucosa en ayunas** y **glicemia** tienen un impacto significativo en la toma de decisiones del modelo, reflejando su enfoque en la evaluación del estado renal y el control glucémico. Además, variables como el **pH** y la **hemoglobina** demostraron ser determinantes en la predicción del deterioro renal. La alta atención asignada a la **infección urinaria** subraya su relevancia en la detección temprana de alteraciones renales.

Estos hallazgos indican que el modelo no solo alcanza una clasificación precisa, sino que también ofrece una interpretación alineada con el conocimiento clínico, fortaleciendo su aplicabilidad en la detección y monitoreo de la nefropatía diabética. Además de representar un avance metodológico en el uso de inteligencia artificial para el diagnóstico médico, este estudio sienta las bases para el desarrollo de sistemas de IA que optimicen la toma de decisiones en hospitales y unidades de salud pública en México, mejorando así la atención y el pronóstico de los pacientes.

Para finalizar, futuras investigaciones podrían incorporar nuevos tipos de datos, para obtener una arquitectura multimodal, el desarrollo de modelos multimodales en salud ha cobrado interés, integrando datos tabulares, imágenes médicas y texto clínico. Sin embargo, la mayoría han sido entrenados con

datos de otras poblaciones, limitando su uso en México. Nuestra arquitectura transformer es un primer paso hacia modelos adaptados a esta población de interés, facilitando la integración de nuevos tipos de datos sin comprometer la precisión diagnóstica. Este trabajo no solo contribuye a la aplicación del AP en la medicina, sino que también abre oportunidades para mejorar la atención de enfermedades crónicas en México.

## Referencias

1. Noor, M.H.M., Ige, A.O.: A Survey on State-of-the-art Deep Learning Applications and Challenges. arXiv print arXiv:2403.17561 (2024). <https://arxiv.org/abs/2403.17561>.
2. Vaswani, A.: Attention Is All You Need. arXiv print arXiv:1706.03762 (2017). <https://arxiv.org/abs/1706.03762>.
3. Ye, H.J. : A Closer Look at Deep Learning on Tabular Data. arXiv print arXiv:2407.00956 (2024). <https://arxiv.org/abs/2407.00956>.
4. Geshkovski, B. : A Mathematical Perspective on Transformers. arXiv print arXiv:2312.10794 (2023). <https://arxiv.org/abs/2312.10794>.
5. Somvanshi, S. : A Survey on Deep Tabular Learning. arXiv print arXiv:2410.12034 (2024). <https://arxiv.org/abs/2410.12034>.
6. Gorishniy, Y., Rubachev, I., Babenko, A.: On Embeddings for Numerical Features in Tabular Deep Learning. *Adv. Neural Inf. Process. Syst.* **35**, 24991–25004 (2022)
7. Zisser, M., Aran, D.: Transformer-based time-to-event prediction for chronic kidney disease deterioration. *J. Am. Med. Inform. Assoc.* **31**(4), 980–990 (2024).
8. Arafat, F. : A Deep Learning Approach to Predict Chronic Kidney Disease in Humans. In: *Proc. IEEE IEMCON*, pp. 1010–1015 (2021)
9. Singh, V., Asari, V.K., Rajasekaran, R.: A Deep Neural Network for Early Detection and Prediction of Chronic Kidney Disease. *Diagnostics* **12**(1), 116 (2022). <https://doi.org/10.3390/diagnostics12010116>
10. Cholakov, R., Kolev, T.: The GatedTabTransformer: An Enhanced Deep Learning Architecture for Tabular Modeling. arXiv print arXiv:2201.00199 (2022). <https://arxiv.org/abs/2201.00199>.
11. Somepalli, G. : SAINT: Improved Neural Networks for Tabular Data via Row Attention and Contrastive Pre-training. arXiv print arXiv:2106.01342 (2021). <https://arxiv.org/abs/2106.01342>.
12. Cai, S., : ARM-Net: Adaptive Relation Modeling Network for Structured Data. In: *Proc. 2021 Int. Conf. Manage. Data*, pp. 207–220 (2021)
13. Huang, X. : TabTransformer: Tabular Data Modeling Using Contextual Embeddings. arXiv print arXiv:2012.06678 (2020). <https://arxiv.org/abs/2012.06678>.
14. Arik, S.Ö., Pfister, T.: TabNet: Attentive Interpretable Tabular Learning. In: *Proc. AAAI Conf. Artif. Intell.* **35**(8), 6679–6687 (2021)
15. Tibshirani, R.: Regression Shrinkage and Selection via the Lasso. *J. R. Stat. Soc. Ser. B* **58**(1), 267–288 (1996)
16. Guyon, I., Weston, J., Barnhill, S., : Gene Selection for Cancer Classification using Support Vector Machines. *Mach. Learn.* **46**(1-3), 389–422 (2002)
17. Base de Datos DiabetIA. <https://repositorio-salud.conacyt.mx/jspui/handle/1000/296>.

18. Scikit-learn developers: InteractiveImputer — scikit-learn 0.24.0 documentation. <https://scikit-learn.org/stable/modules/generated/sklearn.impute.IterativeImputer.html>.
19. Rubin, D.B.: Multiple Imputation for Nonresponse in Surveys. Wiley (1987)