

# UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

## Tecnologías de la Información: Desarrollo y Gestión de Software



### III. Reporte de Métricas de Evaluación

**IDGS91N** - Kevin Iván Aguirre Silva

**Extracción de Conocimiento en Bases de Datos** - Ing.

Luis Enrique Mascote Cano

Chihuahua, Chih., 29 de noviembre de 2025

# Índice

1. Introducción .....	3
2. Investigación de métricas .....	3
2.1. Métricas de evaluación de modelos de clasificación.....	3
<i>Precisión</i> .....	3
<i>AUC (Área bajo la curva ROC)</i> .....	4
<i>Puntuación F1</i> .....	4
<i>Pérdida logarítmica</i> .....	4
2.2. De regresión .....	5
<i>Error Cuadrático Medio (MSE)</i> .....	5
<i>Error Absoluto Medio (MAE)</i> .....	5
3. Solución con KNN.....	6
3.1. Preparación de datos .....	6
<i>Inspección de datos</i> .....	6
<i>División y escalamiento de Datos</i> .....	6
<i>Tamaños de los Conjuntos</i> .....	6
<i>Código</i> .....	7
3.2. Implementación .....	7
<i>Evaluación del Clasificador KNN (F1-score)</i> .....	7
<i>Selección del Mejor Modelo</i> .....	8
<i>Código</i> .....	8
3.3. Evaluación .....	9
<i>Matriz de Confusión</i> .....	9
<i>Métricas de Clasificación</i> .....	9
<i>Curva ROC y AUC</i> .....	10
4. Resultados.....	11
4.1. Comparación del Rendimiento de los Diferentes k.....	11
<i>Análisis de la Comparación</i> .....	11
4.2. Sugerencias y Alternativas de Modelado .....	11
<i>A. Mejoras en el Modelo KNN</i> .....	12
<i>B. Alternativas de Modelado</i> .....	12
<i>C. Consideraciones sobre los Datos</i> .....	12
5. Conclusiones y recomendaciones .....	13
6. Referencias.....	14

## 1. Introducción

El presente informe detalla el proceso de implementación de un modelo de clasificación basado en el algoritmo K-Nearest Neighbors (KNN), aplicado a un conjunto de datos que relaciona las variables glucosa y edad con la variable objetivo etiqueta. El objetivo principal fue desarrollar un clasificador robusto y evaluar su rendimiento utilizando métricas de clasificación estándar. Se llevó a cabo la preparación de los datos mediante la división del conjunto y el escalado de características con StandardScaler. Posteriormente, se entrenó el modelo probando diferentes valores del hiperparámetro k para identificar la configuración óptima basada en el F1-score, seleccionando así el mejor modelo para la evaluación final a través de la matriz de confusión y la curva ROC.

## 2. Investigación de métricas

### 2.1. Métricas de evaluación de modelos de clasificación

#### *Precisión*

Definición: Proporción de predicciones correctas sobre el total de predicciones realizadas.

- **Fórmula:**

$$\frac{TP+TN}{TP+TN+FP+FN}$$

Donde TP = Verdaderos positivos, TN = Verdaderos negativos, FP = Falsos positivos, FN = Falsos negativos.

- **Interpretación:** Indica qué tan a menudo el modelo acierta en sus predicciones.
- **Ventajas:** Fácil de interpretar y funciona bien con clases balanceadas.
- **Limitaciones:** Puede ser engañosa si hay un desequilibrio entre clases, ya que un modelo que siempre predice la clase mayoritaria puede tener alta precisión.

### ***AUC (Área bajo la curva ROC)***

- **Definición:** Mide el área bajo la curva ROC que representa la tasa de verdaderos positivos frente a la tasa de falsos positivos para diferentes umbrales.
- **Fórmula:** Integral de la curva ROC (no tiene fórmula cerrada simple).
- **Interpretación:** Un valor cercano a 1 indica un excelente poder discriminativo, 0.5 indica un modelo sin capacidad predictiva.
- **Ventajas:** Funciona bien para evaluar modelos en conjuntos de datos desequilibrados.
- **Limitaciones:** No refleja la precisión en una probabilidad específica, solo discrimina entre clases.

### ***Puntuación F1***

- **Definición:** Media armónica entre precisión y recuperación (sensibilidad).
- **Formula:**
$$F1 = 2 \times \frac{\text{Precisión} \times \text{Recuperación}}{\text{Precisión} + \text{Recuperación}}$$
- **Interpretación:** Balancea la precisión y la capacidad de encontrar todos los positivos.
- **Ventajas:** Útil cuando se busca equilibrio entre precisión y recuperación.
- **Limitaciones:** No distingue qué error es más costoso y puede no ser intuitiva para algunos casos prácticos.

### ***Pérdida logarítmica***

- **Definición:** Mide la desviación de las probabilidades predichas respecto a las etiquetas reales.
- **Formula:**
$$-\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$
- **Interpretación:** Indica cuán confiables son las probabilidades de las predicciones.
- **Ventajas:** Considera la incertidumbre y penaliza fuertemente las predicciones incorrectas con alta confianza.

- **Limitaciones:** Puede ser difícil de interpretar directamente y es sensible a valores extremos (Microsoft, s.f.).

## 2.2. De regresión

### *Error Cuadrático Medio (MSE)*

- **Definición:** Es el promedio de los cuadrados de las diferencias entre los valores predichos y los reales.
- **Formula:**  

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$
- **Interpretación:** Indica cuánto se desvían en promedio las predicciones del modelo de los valores reales, penalizando más errores grandes.
- **Ventajas:** Penaliza fuertemente errores grandes, útil para modelos donde errores grandes son críticos.
- **Limitaciones:** Sensible a valores atípicos, puede distorsionar la evaluación si hay errores extremos (Maquina, 2022).

### *Error Absoluto Medio (MAE)*

- **Definición:** Promedio de los valores absolutos de las diferencias entre las predicciones y los valores reales.
- **Formula:**  

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$
- **Interpretación:** Mide el error promedio que se espera en las predicciones sin penalizar excesivamente errores grandes.
- **Ventajas:** Fácil de interpretar y menos sensible a valores atípicos que el MSE.
- **Limitaciones:** No penaliza tanto errores grandes como MSE (Dialéktico, 2025).

### 3. Solución con KNN

#### 3.1. Preparación de datos

##### *Inspección de datos*

El archivo Matriz.csv contiene 30 entradas con tres columnas: glucosa, edad y etiqueta. Todas las columnas son de tipo entero (int64) y no tienen valores nulos.

##### *División y escalamiento de Datos*

Se dividió el conjunto de datos, tomando las columnas 'glucosa' y 'edad' como variables predictoras (X) y 'etiqueta' como la variable objetivo (y). Para el escalamiento, se aplicó StandardScaler a las variables predictoras. Este método estandariza las características eliminando la media y escalando a la varianza unitaria (transforma los datos para que tengan una media de  $\mu = 0$  y una desviación estándar de  $\sigma = 1$ , lo cual es crucial para KNN ya que este algoritmo se basa en la distancia entre puntos).

##### *Tamaños de los Conjuntos*

Conjunto	Variables Predictoras (X)	Variable Objetivo (y)
Entrenamiento (70%)	21 filas, 2 columnas	21 filas
Prueba (30%)	9 filas, 2 columnas	9 filas

Primeras Filas de las Variables Predictoras de Entrenamiento Escaladas:

glucosa (escalada)	edad (escalada)
0.380555	0.314664
0.200563	0.153495
-0.627401	-0.249429
-0.339413	-0.007675
0.668542	0.717587

## Código

Librerías e importaciones:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

Cargar y separar los datos:

```
file_path = "Matriz.csv"
df = pd.read_csv(file_path)
```

Variables predictoras (X) y variable objetivo (y):

```
X = df[['glucosa', 'edad']]
y = df['etiqueta']
```

Dividir el conjunto en entrenamiento (70%) y prueba (30%):

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42, stratify=y
)
```

Escalar las variables predictoras (Normalización):

```
scaler = StandardScaler()
```

Ajustar el escalador SÓLO con los datos de entrenamiento y transformar ambos conjuntos:

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

## 3.2. Implementación

### Evaluación del Clasificador KNN (F1-score)

Valor de k	F1-score en el Conjunto de Prueba
3	0.7500
5	0.8571
7	0.8571

## Selección del Mejor Modelo

Ambos valores de  $k = 5$  y  $k = 7$  alcanzaron el F1-score más alto (0.8571). Cuando hay empate en la métrica de selección, se suele elegir el valor de  $k$  más pequeño para reducir la complejidad del modelo (ya que  $k = 5$  requiere menos vecinos que  $k = 7$ ). Por lo tanto, el mejor valor de  $k$  para este caso, basado en el F1-score, es  $k = 5$ .

## Código

Librerías e importaciones:

```
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import f1_score
```

Cargar los datos preparados:

```
X_train_scaled = pd.read_csv("X_train_scaled.csv").values
X_test_scaled = pd.read_csv("X_test_scaled.csv").values
y_train = pd.read_csv("y_train.csv").squeeze()
y_test = pd.read_csv("y_test.csv").squeeze()
```

Definir  $k$  valores a probar:

```
k_values = [3, 5, 7]
results = {}
```

Iterar sobre los valores de  $k$ :

```
for k in k_values:
    # Entrenar el modelo KNN
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train_scaled, y_train)

    # Predecir en el conjunto de prueba
    y_pred = knn.predict(X_test_scaled)

    # Calcular F1-score
    f1 = f1_score(y_test, y_pred, average='binary')
    results[k] = f1
```

Seleccionar el mejor  $k$ :

```
best_k = max(results, key=results.get)
best_f1_score = results[best_k]
```



### 3.3. Evaluación

#### **Matriz de Confusión**

La matriz de confusión es una herramienta esencial que nos muestra el rendimiento del algoritmo en el conjunto de prueba, detallando los aciertos y errores:

$$\text{Matriz de Confusión} = \begin{pmatrix} \text{Verdaderos Negativos (TN)} & \text{Falsos Positivos (FP)} \\ \text{Falsos Negativos (FN)} & \text{Verdaderos Positivos (TP)} \end{pmatrix}$$

Resultados para k = 5:

Valores	Cantidad
Verdaderos Positivos (TP)	3
Verdaderos Negativos (TN)	5
Falsos Positivos (FP)	0
Falsos Negativos (FN)	1

Esto significa que el modelo:

- Acertó en la predicción de 3 casos positivos (TP).
- Acertó en la predicción de 5 casos negativos (TN).
- No cometió errores de tipo I (FP - predijo positivo cuando era negativo).
- Cometió 1 error de tipo II (FN - predijo negativo cuando era positivo).

#### **Métricas de Clasificación**

Con base en la matriz de confusión, se calcularon las métricas de clasificación investigadas:

Métrica	Fórmula	Valor	Interpretación
Precisión General (Accuracy)	$(TP + TN) / (TP + TN + FP + FN)$	0.8889	El 88.98% de todas las predicciones fueron correctas.
Precisión (Precision)	$TP / (TP + FP)$	1.0000	El 100% de los casos que predijo como positivos

			fueron realmente positivos.
Exhaustividad (Recall/TPR)	$TP / (TP + FN)$	0.7500	Logró identificar correctamente el 75.00% e todos los casos positivos reales.
Especificidad (Specificity)	$TN / (TN + FP)$	1.0000	El 100% e los casos negativos reales fueron identificados correctamente.
Puntuación F1 (F1-score)	$2 * (Precision * Recall) / (Precision + Recall)$	0.8571	Es el promedio armónico entre Precisión y Exhaustividad.
AUC	Área bajo la Curva ROC	0.9750	El modelo tiene una alta capacidad para distinguir entre clases positiva y negativa.

### **Curva ROC y AUC**

La Curva Característica Operativa del Receptor (ROC) y el Área Bajo la Curva (AUC) son métricas clave para evaluar la capacidad de discriminación de un modelo.

- AUC mide el rendimiento del modelo en todos los posibles umbrales de clasificación. Un valor de \$1.0\$ representa un modelo perfecto, mientras que \$0.5\$ representa un modelo sin capacidad predictiva (azar).

El modelo KNN ( $k = 5$ ) obtuvo un AUC de 0.9750, lo que indica un rendimiento de clasificación excelente.

El punto de la curva que está más alejado de la línea diagonal de \$0.5\$ representa el mejor rendimiento del modelo, confirmando su alto poder discriminativo.

## 4. Resultados

### 4.1. Comparación del Rendimiento de los Diferentes $k$

La tabla a continuación resume el rendimiento de los tres valores de  $k$  evaluados, utilizando el F1-score como métrica principal:

Valor de $k$	F1-score en el Conjunto de Prueba	Observación
3	0.7500	Rendimiento más bajo; límite de decisión más sensible al ruido.
5	0.8571	Mejor rendimiento; límite de decisión más suave.
7	0.8571	Rendimiento idéntico a $k=5$ , pero con una mayor vecindad.

#### *Análisis de la Comparación*

1.  **$k = 3$  (Baja Regularización):** Un valor de  $k$  pequeño resulta en un modelo con alta varianza (susceptible al ruido o a valores atípicos). El F1-score de 0.7500 sugiere que esta configuración no capturó la relación subyacente tan bien como los valores más grandes.
2.  **$k = 5$  y  $k = 7$  (Mayor Regularización):** Al aumentar el número de vecinos, el modelo promedia más puntos de datos, lo que resulta en un límite de decisión más suave y estable, reduciendo el impacto de los valores atípicos. Ambos alcanzaron el mismo F1-score de 0.8571.

### 4.2. Sugerencias y Alternativas de Modelado

A pesar del excelente desempeño del KNN ( $k=5$  con  $\text{AUC}=0.9750$ ), siempre hay formas de mejorar la robustez y explorar si otros algoritmos son más adecuados.

### **A. Mejoras en el Modelo KNN**

Sugerencia: Búsqueda en Rejilla (GridSearchCV)

Descripción: En lugar de probar solo 3 valores, se puede automatizar la prueba de un rango más amplio de  $k$  (por ejemplo, de 1 a 15) para encontrar el óptimo global de manera sistemática y rigurosa.

### **B. Alternativas de Modelado**

Dado que el problema es de clasificación binaria, existen otros algoritmos potentes que podrían compararse con KNN, especialmente considerando que el KNN no genera una función de decisión explícita (es un algoritmo "perezoso" o lazy learner).

Modelo Alternativo	Ventaja/Uso Sugerido
Regresión Logística	Es un excelente modelo de referencia (baseline). Es simple, rápido de entrenar y proporciona probabilidades fácilmente interpretables.
Máquinas de Soporte Vectorial (SVM)	Muy eficaz en espacios de alta dimensión y cuando hay un límite claro de separación. Con un kernel no lineal (ej. RBF), podría capturar la complejidad de los datos de manera diferente a KNN.
Árboles de Decisión / Random Forest	Los Random Forests (Bosques Aleatorios) son ensambles robustos de árboles de decisión que manejan muy bien las interacciones no lineales entre características y proporcionan una estimación de la importancia de las características.

### **C. Consideraciones sobre los Datos**

El conjunto de datos original (Matriz.csv) es bastante pequeño (30 filas).

- Aumento de Datos: Para un modelado más robusto, se recomienda encarecidamente obtener más datos. Un conjunto de datos tan reducido puede

llevar a que los resultados sean altamente sensibles a la división de entrenamiento/prueba.

- **Análisis de Características:** Aunque solo usamos glucosa y edad, si el archivo original tuviera más características, se podría realizar un análisis de correlación y una selección de características (Feature Selection) para eliminar variables redundantes o poco informativas.

## 5. Conclusiones y recomendaciones

El clasificador K-Nearest Neighbors demostró ser un modelo de clasificación eficaz para este conjunto de datos, logrando un rendimiento óptimo con el hiperparámetro  $k = 5$ . La configuración final del modelo alcanzó un excelente AUC de 0.9750 y una Precisión (Precision) perfecta de 1.0000, aunque la Exhaustividad (Recall) fue de 0.7500. La matriz de confusión reveló que el modelo es altamente selectivo, pues no produjo Falsos Positivos ( $FP=0$ ), priorizando la certeza de las predicciones positivas a costa de clasificar erróneamente un solo caso positivo como negativo ( $FN=1$ ). Estos resultados confirman el alto poder discriminatorio del modelo en la tarea de clasificación.

Para aumentar la robustez y la Exhaustividad del modelo, se recomienda enfáticamente obtener y utilizar un conjunto de datos más grande que mitigue la alta sensibilidad del modelo a un tamaño muestral de solo 30 filas. Además, se sugiere implementar la Validación Cruzada junto con una búsqueda en rejilla (GridSearchCV) para probar un rango más amplio de valores de  $k$  y sus ponderaciones, asegurando una selección de hiperparámetros más rigurosa. Finalmente, se aconseja comparar el KNN con alternativas robustas como Random Forest o Máquinas de Soporte Vectorial (SVM) para verificar si un modelo no lineal distinto puede mejorar el equilibrio entre Precisión y Exhaustividad.

## 6. Referencias

- Dialéktico, D. (14 de Julio de 2025). *Métricas de evaluación de modelos de regresión*.  
Obtenido de Dialéktico: <https://dialektico.com/metricas-de-evaluacion-de-modelos-de-regresion/>
- Maquina, C. (4 de Enero de 2022). *Las Mejores Métricas para Evaluar Modelos de Regresión con Scikit-Learn: R2, MSE, RMSE, MAE y otras*. Obtenido de YouTube: <https://www.youtube.com/watch?v=9IZ6OPQWtpw>
- Microsoft. (s.f.). *Evaluación del modelo de ML.NET con métricas*. Obtenido de Microsoft Learn: <https://learn.microsoft.com/es-es/dotnet/machine-learning/resources/metrics>