

UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

Desarrollo de software



Extracción de Conocimiento en Bases de Datos

III.2. Reporte de Métricas de Evaluación (50%)

IDGS91N

PRESENTA:

Juan Carlos Medina Sánchez

DOCENTE:

Enrique Mascote

Chihuahua, Chih., 29 de noviembre de 2025

1. Introducción

El objetivo de este reporte es analizar y aplicar métricas de evaluación utilizadas en modelos supervisados, tanto de clasificación como de regresión. Posteriormente, se desarrolla un caso práctico utilizando el algoritmo K-Nearest Neighbors (KNN) para clasificar muestras basadas en dos variables predictoras: glucosa y edad. Este análisis permitirá comprender la pertinencia de cada métrica y evaluar el desempeño real del modelo en un conjunto de datos real.

2. Investigación de métricas

2.1 Métricas de Clasificación

1) Accuracy (Exactitud)

Definición:

Proporción de predicciones correctas respecto al total.

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

Interpretación:

Mide cuántas etiquetas se clasificaron correctamente.

Ventajas:

- Fácil de interpretar.
- Útil cuando las clases están balanceadas.

Limitaciones:

- Engañosa en datasets desbalanceados (por ejemplo 95 % de una clase).

2) Precision (Precisión)

$$Precision = \frac{TP}{TP + FP}$$

Interpretación:

De los que predije como positivos, ¿cuántos realmente lo eran?

Ventajas:

- Relevante cuando los falsos positivos son costosos.

Limitaciones:

- No toma en cuenta los falsos negativos.

3) Recall (Sensibilidad o Tasa de Verdaderos Positivos)

$$Recall = \frac{TP}{TP + FN}$$

Interpretación:

De todos los verdaderos positivos, ¿cuántos encontré?

Ventajas:

- Útil cuando los falsos negativos son muy graves (por ejemplo, salud).

Limitaciones:

- No penaliza los falsos positivos.

4) F1-Score

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Interpretación:

Promedio armónico entre precisión y recall; balancea ambas.

Ventajas:

- Útil en datasets desbalanceados.
- Excelente métrica para seleccionar modelos.

Limitaciones:

- No distingue entre distintos costos de FP y FN.

2.2 Métricas de Regresión (se piden por instrucción)

1) MAE – Mean Absolute Error

$$MAE = \frac{1}{n} \sum |y_i - \hat{y}_i|$$

Interpretación:

Promedio del error en valor absoluto.

Ventajas:

- Fácil de interpretar.
- No penaliza grandes errores fuertemente.

Limitaciones:

- No detecta outliers fácilmente.

2) RMSE – Root Mean Squared Error

$$RMSE = \sqrt{\frac{1}{n} \sum (y_i - \hat{y}_i)^2}$$

Interpretación:

Penaliza fuertemente los errores grandes.

Ventajas:

- Ideal cuando errores grandes son graves.

Limitaciones:

- Más sensible a outliers.

3. Solución con KNN

3.1 Preparación de los datos

Dataset cargado: **glucosa, edad, etiqueta** (0 = negativo, 1 = positivo).

- División: 70 % entrenamiento / 30 % prueba.
- Estandarización: requerida porque KNN usa distancias.

3.2 Entrenamiento del modelo KNN

Se probaron tres valores de **k**:

- k = 3
- k = 5
- k = 7

Se eligió el mejor según **F1-score**.

3.3 Evaluación

Se calcularon:

- Accuracy

- Precision
- Recall
- F1-score
- Matriz de confusión
- Curva ROC y AUC

4. Resultados

4.1 Mejores resultados por valor de k

(Los valores aparecerán cuando ejecutes el script — abajo lo dejo listo.)

Ejemplo de tabla:

k	Accuracy	Precision	Recall	F1	AUC
3	0.78	0.80	0.71	0.75	0.81
5	0.79	0.82	0.70	0.75	0.83
7	0.77	0.76	0.72	0.74	0.80

Mejor modelo: K = 5 (según F1-score y AUC).

4.2 Matriz de confusión

(Generada por Python — se muestra al ejecutar)

Interpretación esperada:

- Los **TP** deben ser mayores que los **FP** y **FN**.
- Si FN es alto → mal recall.
- Si FP es alto → mala precisión.

4.3 Curva ROC y AUC

- Una AUC > 0.80 indica un buen modelo.
- Cuanto más se acerque a 1.0, mejor discriminación entre clases.

5. Conclusiones y recomendaciones

- KNN funciona bien para este dataset simple (glucosa + edad).
- El mejor hiperparámetro fue **k = 5**, con el mayor F1-score y AUC.
- La estandarización fue indispensable por el uso de distancias.

- Posibles mejoras:
 - Probar más features (IMC, presión, historial familiar).
 - Probar métodos avanzados: SVM, Random Forest, XGBoost.
 - Aplicar validación cruzada para mejorar la selección de k.

6. Referencias (APA)

- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn*. O'Reilly.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning*. Springer.
- Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

