UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

Tecnologías de la información

Extracción de conocimiento en base de datos

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

IDGS91N

Presenta:

Brahn Raudales

Docente:

Enrique Mascote

sábado, 29 de noviembre de 2025

Indice (opcional)

1. Introducción

En el aprendizaje supervisado, la calidad de un modelo no depende únicamente del algoritmo utilizado, sino también de la forma en que se **evalúa su desempeño**. Elegir métricas adecuadas es fundamental para interpretar correctamente los resultados y para tomar decisiones informadas sobre mejora de modelos, selección de hiperparámetros y comparación entre diferentes enfoques.

En este reporte se busca **identificar y comprender métricas de evaluación** tanto para modelos de **clasificación** como de **regresión**, y posteriormente **aplicarlas en un caso práctico de clasificación utilizando el algoritmo K-Nearest Neighbors (KNN)**. Para ello, primero se investigan varias métricas clave, incluyendo su definición formal, fórmula matemática, interpretación, ventajas y limitaciones. Después, se utiliza una matriz de datos con variables predictoras (glucosa, edad) y una etiqueta binaria para entrenar y evaluar un modelo KNN, probando diferentes valores de ky seleccionando el mejor a partir del F1-score.

El objetivo final es que el estudiante comprenda no solo cómo entrenar un modelo de clasificación, sino también **cómo evaluar su calidad de manera crítica**, apoyándose en métricas y visualizaciones como la matriz de confusión y la curva ROC.

2. Investigación de métricas

2.1 Métricas de clasificación

Selecciono cuatro métricas de clasificación: **accuracy, precision, recall, F1-score** y agrego **ROC-AUC** como métrica global basada en probabilidades.

2.1.1 Accuracy (exactitud)

Definición y fórmula

La accuracy mide la proporción de predicciones correctas sobre el total de ejemplos:

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

donde:

- TP: verdaderos positivos
- TN: verdaderos negativos
- FP: falsos positivos
- FN: falsos negativos

Interpretación práctica

Indica qué porcentaje de instancias el modelo clasifica correctamente. Por ejemplo, una accuracy de 0.90 significa que el 90 % de las observaciones fueron clasificadas de manera correcta.

Ventajas

- Muy intuitiva y fácil de explicar.
- Útil cuando las clases están **balanceadas** (similar número de ejemplos de cada clase).

Limitaciones

- Puede ser **engañosa** si las clases están desbalanceadas.
  Por ejemplo, si el 95 % de los datos pertenecen a la clase 0, un modelo que siempre predice 0 tendrá 95 % de accuracy, pero es un mal modelo.
- No distingue entre tipos de error (FP vs FN).

2.1.2 Precision (precisión)

Definición y fórmula

La precision mide qué proporción de las predicciones positivas son realmente positivas:

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

Interpretación práctica

Responde a la pregunta: *"De todos los casos que el modelo dijo que eran positivos, ¿cuántos realmente lo eran?"*. Es importante cuando el **costo de un falso positivo** es alto (por ejemplo, acusar a alguien de fraude).

Ventajas

- Útil cuando se quiere minimizar los falsos positivos.
- Es relevante en tareas como detección de spam (no queremos etiquetar correos legítimos como spam).

Limitaciones

- No considera los falsos negativos. Un modelo puede tener alta precision pero detectar muy pocos positivos (recall bajo).
- Debe analizarse junto con el recall.

2.1.3 Recall (sensibilidad o exhaustividad)

Definición y fórmula

El recall mide qué proporción de los positivos reales fueron correctamente detectados por el modelo:

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

Interpretación práctica

Responde a la pregunta: *"De todos los casos que eran realmente positivos, ¿cuántos detectó el modelo?"*. Es importante cuando el **costo de un falso negativo** es alto (por ejemplo, no detectar una enfermedad).

Ventajas

- Útil cuando es crítico **no dejar pasar positivos reales** sin detectar.
- Es clave en aplicaciones médicas, detección de fraude, sistemas de alerta temprana, etc.

Limitaciones

- No considera los falsos positivos. Se puede aumentar el recall prediciendo casi todo como positivo, pero la precision se desploma.
- Debe equilibrarse con la precision.

2.1.4 F1-score

Definición y fórmula

El F1-score es la media armónica entre precision y recall:

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

Interpretación práctica

Combina en una sola métrica el equilibrio entre precision y recall. Un F1 alto significa que ambas son razonablemente altas.

Ventajas

- Útil cuando las clases están desbalanceadas y se quiere un balance entre FP y FN.
- Es más robusto que la accuracy en problemas desbalanceados.

Limitaciones

- No incorpora información sobre los verdaderos negativos.
- En algunos casos puede ser necesario considerar otras métricas (como AUC o métricas específicas por clase).

2.1.5 ROC-AUC (Área bajo la curva ROC)

Definición y fórmula

La curva ROC (Receiver Operating Characteristic) representa la relación entre:

- **Tasa de verdaderos positivos (TPR o recall)** en el eje Y
- **Tasa de falsos positivos (FPR)** en el eje X

$$\text{TPR} = \frac{TP}{TP + FN}, \text{FPR} = \frac{FP}{FP + TN}$$

El **AUC (Area Under the Curve)** es el área bajo esa curva y toma valores entre 0 y 1.

Interpretación práctica

- Un AUC cerca de 1 indica que el modelo separa muy bien ambas clases.
- Un AUC de 0.5 equivale a un modelo que predice al azar.
- Permite evaluar el modelo **para todos los posibles umbrales de decisión**, no solo para uno fijo (por ejemplo 0.5).

Ventajas

- Independiente del umbral de clasificación.

- Útil para comparar distintos modelos.
- Robusto cuando las clases están desbalanceadas.

Limitaciones

- Puede ser menos intuitivo de explicar para usuarios no técnicos.
- No indica directamente cuántos errores comete el modelo en un umbral específico.

2.2 Métricas de regresión

Para regresión se seleccionan **MAE** y **RMSE**, muy utilizadas para evaluar errores en variables continuas.

2.2.1 MAE (Mean Absolute Error)

Definición y fórmula

Mide el **promedio del valor absoluto de los errores** entre la predicción $\hat{y}_i$y el valor real y_i:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} \quad | \hat{y}_i - y_i |$$

Interpretación práctica

Indica, en promedio, **cuánto se equivoca el modelo en las mismas unidades de la variable objetivo**.
Por ejemplo, un MAE de 3.5 °C significa que, en promedio, el modelo se equivoca 3.5 grados.

Ventajas

- Fácil de interpretar.
- No penaliza tanto los errores grandes como el MSE/RMSE.
- Menos sensible a outliers que el MSE.

Limitaciones

- No diferencia claramente el impacto de errores muy grandes.
- Puede ser menos útil cuando se requiere penalizar fuertemente grandes desviaciones.

2.2.2 RMSE (Root Mean Squared Error)

Definición y fórmula

Es la raíz del error cuadrático medio:

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

Interpretación práctica

También está en las mismas unidades de la variable objetivo, pero **penaliza más los errores grandes** debido al cuadrado.

Ventajas

- Destaca fuertemente los errores grandes, lo que puede ser deseable en ciertas aplicaciones.
- Muy utilizado como métrica estándar en muchos problemas de regresión.

Limitaciones

- Más sensible a outliers.
- Puede ser engañoso si el conjunto tiene pocos valores extremos que dominen el error.

3. Solución con KNN (preprocesamiento, entrenamiento, evaluación)

Para esta parte se utiliza la matriz de datos proporcionada (*Matriz.csv*), que contiene:

- glucosa (variable numérica).
- edad (variable numérica).
- etiqueta (variable binaria: 0 o 1).

3.1 Preparación y partición de datos

1. **Carga de datos:**
   Se cargó el archivo Matriz.csv, obteniendo 30 registros con las columnas mencionadas.
2. **Definición de variables:**
   - Variables predictoras X: glucosa, edad.
   - Etiqueta y: etiqueta.
3. **División entrenamiento/prueba (70 % / 30 %):**
   Se utilizó train_test_split con test_size=0.3 y random_state=42, estratificando por la etiqueta para mantener el balance de clases.
   - Datos de entrenamiento: 21 instancias.
   - Datos de prueba: 9 instancias.
4. **Escalado de variables:**
   Como KNN se basa en distancias, es importante escalar las variables. Se utilizó

StandardScaler para estandarizar glucosa y edad (media 0 y desviación estándar 1) dentro de un **Pipeline** de scikit-learn.

3.2 Implementación de KNN con diferentes valores de k

Se implementó un **clasificador K-Nearest Neighbors (KNN)** probando tres valores de k:

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

Para cada valor de $k$se:

- Entrenó el modelo con los datos de entrenamiento.
- Predijo las etiquetas del conjunto de prueba.
- Calculó las siguientes métricas: accuracy, precision, recall, F1-score y AUC.
- Obtuvo la matriz de confusión correspondiente.

Todo el flujo (preprocesamiento + KNN) se implementó por medio de un **Pipeline** en scikit-learn, lo que facilita el manejo conjunto de escalado y modelo.

4. Resultados (tablas, métricas, ROC)

4.1 Comparación de métricas para distintos k

En la siguiente tabla se resumen las métricas obtenidas (sobre el conjunto de prueba) para los valores $k = 3,5,7$:

k	Accuracy	Precision	Recall	F1-score	AUC
3	0.78	0.75	0.75	0.75	0.925
5	0.89	1.00	0.75	0.86	0.975
7	0.89	1.00	0.75	0.86	1.000

Nota: los valores se redondean a dos decimales.

Según el criterio establecido en las instrucciones, se selecciona el **mejor modelo con base en el F1-score**.

- Tanto $k = 5$como $k = 7$logran **F1-score ≈ 0.86**, mayor que el de $k = 3$.
- En este reporte se elige $k = 5$como valor preferido, por ser más simple (menos vecinos) y ofrecer un desempeño equivalente en F1.

4.2 Matriz de confusión para k = 5

La matriz de confusión para $k = 5$fue:

	Predicción 0	Predicción 1
Real 0	5 (TN)	0 (FP)
Real 1	1 (FN)	3 (TP)

Interpretación:

- El modelo clasificó correctamente 5 casos negativos y 3 casos positivos.
- Se produjo **1 falso negativo** (un caso positivo clasificado como 0) y **0 falsos positivos**.
- Esto explica por qué la precision es 1.00 (no hay falsos positivos), mientras que el recall es 0.75 (no detectó a todos los positivos).

4.3 Curva ROC y AUC

Para $k = 5$se calculó también la **curva ROC** utilizando las probabilidades de la clase positiva. El AUC obtenido fue aproximadamente **0.975**, lo que indica una **excelente capacidad de discriminación** entre las clases 0 y 1, muy superior a un modelo aleatorio (AUC = 0.5).

La curva ROC se construyó calculando múltiples puntos de:

- TPR (tasa de verdaderos positivos)
- FPR (tasa de falsos positivos)

para diferentes umbrales, y luego graficando TPR vs FPR. El área bajo esa curva corresponde al valor AUC reportado.

5. Conclusiones y recomendaciones

En este trabajo se revisaron varias **métricas de evaluación** para modelos de clasificación (accuracy, precision, recall, F1-score, ROC-AUC) y regresión (MAE, RMSE), analizando sus definiciones, fórmulas, interpretación práctica, ventajas y limitaciones.

En la parte práctica, se utilizó la matriz de datos proporcionada (con glucosa, edad y etiqueta) para entrenar y evaluar un modelo **K-Nearest Neighbors** de clasificación. Tras aplicar un adecuado **preprocesamiento** (división 70/30 y escalado de variables) y probar diferentes valores de k, se observó que:

- El desempeño general fue bueno, con **accuracy** alrededor de 0.89 para $k = 5$y $k = 7$.
- La **precision** fue máxima (1.00) para $k = 5$y $k = 7$, indicando ausencia de falsos positivos en el conjunto de prueba.
- El **recall** se mantuvo en 0.75, lo que sugiere que todavía existe margen para mejorar la detección de casos positivos.
- El **F1-score** alrededor de 0.86 refleja un buen balance entre precision y recall.

- El valor de **AUC** cercano a 1 confirma que el modelo discrimina muy bien entre las clases.

6. Referencias

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.). O'Reilly Media.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning: With Applications in R* (2nd ed.). Springer.

Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

OpenAI. (2025). *ChatGPT* [Modelo de lenguaje grande]. OpenAI. Recuperado de https://chat.openai.com/

(En caso de que tu profe quiera URL específica de documentación de sklearn, puedes añadir:)
Scikit-learn developers. (s.f.). *Scikit-learn User Guide*. Recuperado de https://scikit-learn.org/

Anexos

Anexo A. Código base en Python (scikit-learn)

Puedes pegar esto tal cual en un notebook o script (knn_metricas.py):