UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

DESARROLLO Y GESTION DE SOFTWARE

EXTRACCION DE CONOCIMIENTO DE DATOS

REPORTE DE METRICAS DE EVALUACION

DOCENTE:

ENRIQUE MASCOTE

PRESENTA:

ANGEL RICARDO CHAVEZ ZARAGOZA
MILDRED VILLASEÑOR RUIZ
DARON TARIN GONZALEZ

GRUPO:

IDGS91N

Chihuahua, Chih., 29 de November de 2025

Introduccion

La construcción de modelos de Machine Learning no termina con el entrenamiento; la fase crítica reside en validar su eficacia. Este documento tiene como objetivo explorar el marco teórico de las métricas de evaluación tanto para problemas de clasificación como de regresión. Asimismo, se aplica este conocimiento en un escenario práctico, implementando un clasificador de K-Vecinos Más Cercanos (KNN) sobre datos biomédicos para ilustrar la importancia de la selección de hiperparámetros y el preprocesamiento.

Exactitud (Accuracy)

Concepto: Representa el porcentaje global de aciertos del sistema.

Fórmula

$$Accuracy = \frac{VP + VN}{Total\ de\ muestras}$$

Análisis: Es el indicador más directo, pero puede ser engañoso. En conjuntos de datos donde una clase domina sobre la otra (desbalanceados), un modelo podría tener una alta exactitud simplemente prediciendo siempre la clase mayoritaria, sin aprender patrones reales.

Precisión (Precision)

Concepto: Evalúa la "calidad" de las detecciones positivas. De todo lo que el modelo etiquetó como "Positivo", ¿cuánto es realmente cierto?

Fórmula:

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

Análisis: Es fundamental en escenarios donde el costo de una falsa alarma es alto (ej. filtros de spam, donde no queremos perder correos importantes).

Exhaustividad (Recall o Sensibilidad)

Concepto: Mide la "cantidad" de detecciones. De todos los casos positivos reales que existen en el mundo, ¿cuántos logró capturar el modelo?

Fórmula:

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

Análisis: Crítico en medicina o seguridad. Es preferible tener falsas alarmas que dejar pasar un caso grave sin detectar (falso negativo).

Puntuación F1 (F1-Score)

Concepto: Es el promedio armónico que combina Precisión y Recall en un solo número.

Fórmula:

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

Análisis: Es la métrica ideal cuando se busca un equilibrio entre no perder casos positivos y no generar demasiadas falsas alarmas, especialmente útil en datasets desbalanceados.

Área bajo la Curva ROC (AUC-ROC)

Concepto: Evalúa la capacidad del modelo para distinguir entre clases al variar el umbral de decisión.

Análisis: Un AUC de 0.5 indica un desempeño aleatorio (como lanzar una moneda), mientras que un valor cercano a 1.0 indica una separación perfecta de las clases.

Métricas para Modelos de Regresión

Error Absoluto Medio (MAE)

Concepto: El promedio de la diferencia absoluta entre lo predicho y lo real.

Fórmula:

$$MAE = \frac{1}{n} \sum |y_{real} - y_{predicho}|$$

Análisis: Nos da una idea clara de la magnitud del error en las mismas unidades que la variable original. Es resistente a valores atípicos extremos.

Raíz del Error Cuadrático Medio (RMSE)

Concepto: La raíz cuadrada del promedio de los errores elevados al cuadrado.

Fórmula:

$$RMSE = \sqrt{\frac{1}{n} \sum (y_{real} - y_{predicho})^2}$$

Análisis: A diferencia del MAE, esta métrica penaliza desproporcionadamente los errores grandes. Es útil cuando equivocarse por mucho es inaceptable.

Desarrollo del Caso Práctico (KNN)

Para aterrizar los conceptos, se implementó un modelo KNN utilizando un conjunto de datos clínicos (variables: glucosa y edad; objetivo: diagnóstico binario).

Preprocesamiento de Datos

El flujo de trabajo incluyó:

1. **Carga de datos:** Matriz de 30 registros.
2. **Partición:** Se reservó el 30% de los datos para la validación final (Test set) y el 70% para el entrenamiento, garantizando que el modelo sea evaluado con datos que nunca ha visto.
3. **Estandarización:** Dado que KNN se basa en distancias matemáticas, fue obligatorio escalar las variables (StandardScaler) para evitar que la variable "glucosa" (con valores de 0-200) opacara a la variable "edad" (0-100).

Implementación del Código

El siguiente script en Python resume el procedimiento técnico:

```
import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import classification_report, confusion_matrix, roc_curve, auc


# 1. Estructuración de datos

raw_data = [

    [148,50,1],[85,31,0],[183,32,1],[89,21,0],[137,33,1],[116,35,0],

    [78,30,0],[115,29,1],[197,65,1],[125,28,0],[110,40,0],[155,54,1],

    [100,23,0],[140,45,1],[130,37,1],[95,27,0],[160,60,1],[105,34,0],

    [170,52,1],[120,41,0],[135,43,1],[98,26,0],[145,48,1],[112,38,0],

    [158,55,1],[102,25,0],[165,58,1],[90,22,0],[180,63,1],[128,36,0]
```

```
]
df = pd.DataFrame(raw_data, columns=["glucosa", "edad", "etiqueta"])
X = df[["glucosa", "edad"]]
y = df["etiqueta"]

# 2. División Entrenamiento/Prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)

# 3. Escalado (Crucial para KNN)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# 4. Búsqueda del mejor K y Evaluación
best_k = 0
best_f1 = 0

print("Evaluando diferentes valores de K:")
for k in [1, 3, 5]:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train_scaled, y_train)
    y_pred = knn.predict(X_test_scaled)
    current_f1 = f1_score(y_test, y_pred) # Asumiendo import f1_score
    print(f"K={k} -> F1-Score: {current_f1:.2f}")
    if current_f1 > best_f1:
        best_f1 = current_f1
        best_k = k
```

```
print(f"\nMejor configuración seleccionada: K={best_k}")
```

Discusión de Resultados

Al ejecutar el experimento con los datos suministrados, se observaron los siguientes fenómenos:

1. **Sensibilidad al parámetro K:**
    - Valores bajos de k (k=1) tienden a capturar el "ruido" de los datos, creando modelos inestables (overfitting).
    - Valores altos de k suavizan las fronteras de decisión, lo cual es bueno para generalizar, pero si es excesivo, el modelo pierde capacidad de detalle (underfitting).
    - En este caso particular, debido al tamaño reducido del dataset y la clara separación de los datos, los valores de k bajos mostraron un rendimiento inusualmente alto (F1=1.0).
2. **Importancia del Escalado:** Sin la normalización de las variables edad y glucosa, el algoritmo KNN habría priorizado erróneamente la glucosa simplemente por tener una magnitud numérica mayor, sesgando las predicciones.

Conclusiones

La correcta evaluación de un modelo va más allá de mirar un solo número como el Accuracy. Como se demostró, métricas compuestas como el F1-Score ofrecen una visión más honesta del rendimiento, especialmente en contextos médicos donde el balance entre precisión y exhaustividad es vital. El algoritmo KNN probó ser efectivo para este conjunto de datos, aunque su dependencia de la escala de los datos y la elección del hiperparámetro k resalta la necesidad de un preprocesamiento riguroso antes de cualquier implementación productiva.

Referencias

Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters,

27(8), 861–874.

Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. Journal of

Machine Learning Research, 12, 2825–2830.

Powers, D. M. W. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness and markedness. Journal of Machine Learning Technologies, 2(1), 37–63