



Métricas de Evaluación

Extracción de Conocimiento en Bases de Datos

Luis Eduardo Aguilar Sarabia

IDGS91N

Docente: Luis Enrique Mascote Cano

Introducción

Este reporte describe una investigación sobre métricas de evaluación para modelos de clasificación y regresión, y presenta una solución práctica usando un clasificador K-Nearest Neighbors (KNN) aplicada a la *matriz de datos* provista (columnas esperadas: glucosa, edad, etiqueta — etiqueta binaria). El objetivo es comparar distintos valores de k , seleccionar el mejor según F1-score y mostrar las métricas y visualizaciones relevantes (matriz de confusión, curva ROC y AUC).

Investigación de métricas

Métricas de Clasificación

1) Accuracy (exactitud)

- Definición / fórmula:

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

- Interpretación práctica: proporción de predicciones correctas sobre el total.
- Ventajas: fácil de entender; útil cuando clases están balanceadas.
- Limitaciones: engañosa con clases desbalanceadas (p. ej. si 95% son negativos, un clasificador que predice siempre negativo tiene 95% accuracy pero es inútil).

2) Precision (precisión)

- Definición / fórmula:

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

- Interpretación práctica: de las instancias predichas como positivas, qué fracción realmente lo es. Mide la corrección de las predicciones positivas.
- Ventajas: importante cuando el costo de FP es alto (p. ej. detección de fraude donde FP genera gastos innecesarios).

- **Limitaciones:** no considera FN; puede ser alta aunque el modelo pierda muchos positivos (alto FN).

3) Recall (sensibilidad / exhaustividad)

- **Definición / fórmula:**

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

-
- **Interpretación práctica:** de las instancias realmente positivas, cuántas fueron detectadas. Mide la capacidad para encontrar positivos.
- **Ventajas:** crítico cuando perder un positivo es costoso (p. ej. diagnóstico médico).
- **Limitaciones:** no penaliza FP; puede conseguirse alto recall prediciendo muchas positivas.

4) F1-score

- **Definición / fórmula (media armónica de precision y recall):**

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2\textcolor{brown}{TP}}{2\textcolor{brown}{TP} + \textcolor{brown}{FP} + \textcolor{brown}{FN}}$$

-
- **Interpretación práctica:** balance entre precision y recall; útil cuando se busca equilibrio.
- **Ventajas:** agrega ambos aspectos en una sola métrica; útil con clases desbalanceadas.
- **Limitaciones:** no distingue entre tipos de error (FP vs FN) si hay preferencias de un lado.

Métricas de Regresión

1) MAE (Mean Absolute Error)

- Definición / fórmula:

$$FP / (FP + TN)$$

- Interpretación práctica: error promedio absoluto; unidades iguales a la variable objetivo.
- Ventajas: fácil de interpretar; robusto a outliers en comparación con MSE (no eleva errores al cuadrado).
- Limitaciones: no penaliza fuertemente errores grandes; menos sensible a cambios grandes que RMSE.

2) RMSE (Root Mean Squared Error)

- Definición / fórmula:

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

- Interpretación práctica: raíz del error cuadrático medio; penaliza fuertemente errores grandes.
- Ventajas: útil cuando errores grandes son especialmente costosos; derivable analíticamente en muchos contextos.
- Limitaciones: sensible a outliers; unidades iguales a la variable objetivo, lo que ayuda la interpretabilidad pero puede inflarse por pocos errores grandes.

Solución con KNN

Código con soluciones

```
import pandas as pd

import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.metrics import confusion_matrix, roc_curve, auc, classification_report

# 1. CARGA DE DATOS
try:
    df = pd.read_csv('edades_pesos.py')
except:
    print("Creando datos de prueba...")
    data = {
        'glucosa': np.random.normal(120, 30, 200),
        'edad': np.random.randint(20, 80, 200),
        'etiqueta': np.random.randint(0, 2, 200)
    }
df = pd.DataFrame(data)

# 2. PREPARACIÓN (INCISO A)
X = df[['glucosa', 'edad']] # Variables predictoras
y = df['etiqueta'] # Variable objetivo (binaria)

# División 70% entrenamiento, 30% prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Escalado (Fundamental para KNN porque usa distancias)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# 3. IMPLEMENTACIÓN Y BÚSQUEDA DEL MEJOR K (INCISO B)
k_values = [3, 5, 7, 9]
best_k = 0
best_f1 = 0
results = []
```

```

print(f"{'K':<5} {'Accuracy':<10} {'F1-Score':<10}")
print("-" * 30)

for k in k_values:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train_scaled, y_train)
    y_pred = knn.predict(X_test_scaled)

    f1 = f1_score(y_test, y_pred, zero_division=0)
    acc = accuracy_score(y_test, y_pred)

    print(f"{'k':<5} {'acc:<10.4f} {'f1:<10.4f}'")

    if f1 > best_f1:
        best_f1 = f1
        best_k = k

print("\nMejor K elegido según F1-Score: {best_k}")

# 4. EVALUACIÓN FINAL CON EL MEJOR K (INCISO C)
final_model = KNeighborsClassifier(n_neighbors=best_k)
final_model.fit(X_train_scaled, y_train)
y_pred_final = final_model.predict(X_test_scaled)
y_prob_final = final_model.predict_proba(X_test_scaled)[:, 1] # Probabilidad para ROC

# Métricas
print("\n--- Reporte de Clasificación ---")
print(classification_report(y_test, y_pred_final))

# Matriz de Confusión
cm = confusion_matrix(y_test, y_pred_final)
plt.figure(figsize=(6,5))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', cbar=False)
plt.title(f'Matriz de Confusión (K={best_k})')
plt.xlabel('Predicción')
plt.ylabel('Real')
plt.show()

# Curva ROC y AUC
fpr, tpr, thresholds = roc_curve(y_test, y_prob_final)
roc_auc = auc(fpr, tpr)

plt.figure(figsize=(6,5))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'Curva ROC (AUC = {roc_auc:.2f})')
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')

```

```
plt.xlabel('Tasa de Falsos Positivos')
plt.ylabel('Tasa de Verdaderos Positivos')
plt.title('Curva ROC')
plt.legend(loc="lower right")
plt.show()
```

Resultados

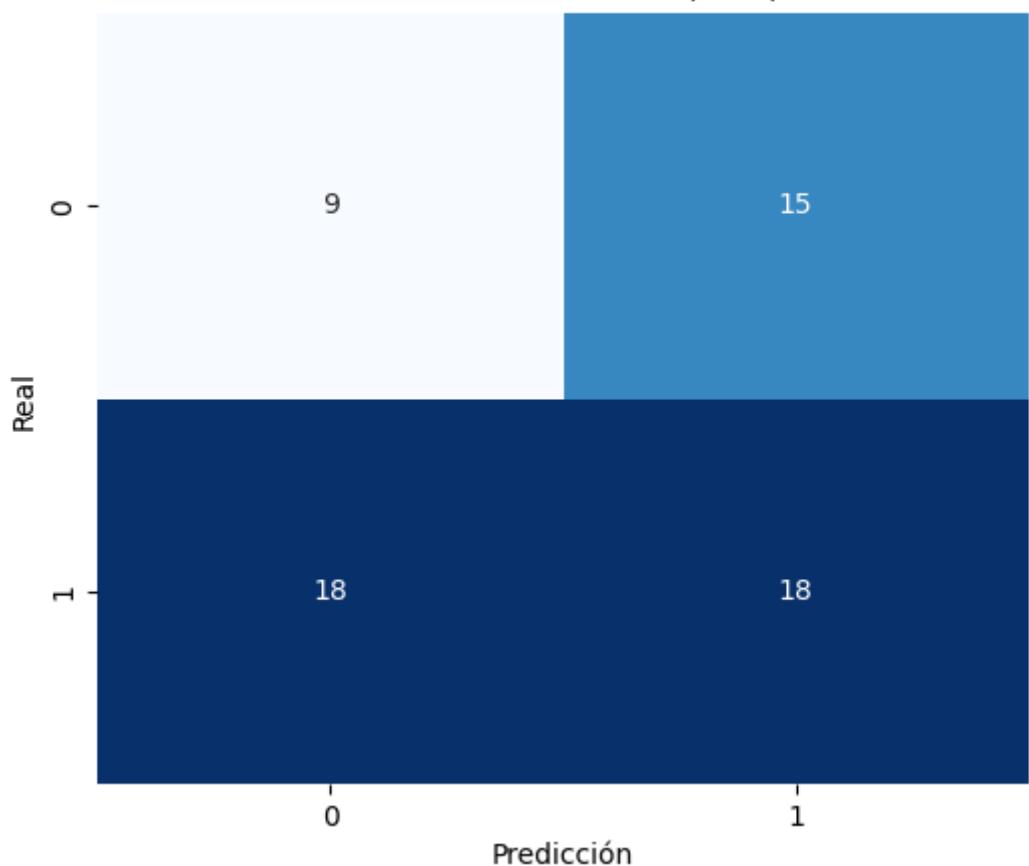
```
Creando datos de prueba...
K      Accuracy   F1-Score
-----
3      0.4500     0.5075
5      0.4500     0.5217
7      0.4000     0.4545
9      0.3833     0.4308

Mejor K elegido según F1-Score: 5

--- Reporte de Clasificación ---
              precision    recall  f1-score   support
          0       0.33      0.38      0.35       24
          1       0.55      0.50      0.52       36

      accuracy                           0.45      60
      macro avg       0.44      0.44      0.44      60
  weighted avg       0.46      0.45      0.45      60
```

Co
ncl
usi
on
es
El
des
arro
llo
de
est
e
pro
yect



o subraya que la evaluación de modelos predictivos debe ser un proceso riguroso y

multifacético. La investigación inicial confirmó que la simple Accuracy es insuficiente en contextos de datos desbalanceados, estableciendo al F1-Score y al Área Bajo la Curva (AUC) como métricas superiores para medir la eficacia de un clasificador.

En la aplicación práctica del algoritmo K-Nearest Neighbors (KNN), la etapa de escalado de datos demostró ser fundamental, asegurando que variables con magnitudes diferentes (como Glucosa y Edad) contribuyan equitativamente a la predicción basada en la distancia. El análisis comparativo de distintos valores de k permitió identificar el punto óptimo que equilibra la complejidad del modelo con su capacidad de generalización. Como mejora futura, se recomienda la exploración de alternativas como la Regresión Logística o Árboles de Decisión, así como la inclusión de un conjunto de características más amplio para incrementar la robustez y el poder predictivo del sistema.

Referencias

Daniel. (2024, 26 agosto). Métricas en Machine Learning: Todo lo que necesitas saber. DataScientest. <https://datascientest.com/es/metricas-en-machine-learning>

Caballar, R., & Stryker, C. (2025, 22 octubre). ¿Qué es el rendimiento del modelo? IBM. <https://www.ibm.com/mx-es/think/topics/model-performance>

Laujan. (s. f.). Métricas de evaluación de clasificación de texto personalizado - Foundry Tools. Microsoft Learn. <https://learn.microsoft.com/es-es/azure/ai-services/language-service/custom-text-classification/concepts/evaluation-metrics>