



EXTRACCIÓN DE CONOCIMIEN TOS EN BASES DE DATOS

ING. LUIS ENRIQUE MASCOTE CANO



MÉTRICAS DE EVALUACIÓN
Lic. Ricardo
Hernández Martínez
Fecha de Entrega:
29/11/25

Introducción

La evaluación de modelos de clasificación y regresión es fundamental para medir su desempeño y capacidad de generalización. En este documento se investigan métricas esenciales para ambos tipos de modelos, describiendo su definición formal, interpretación y limitaciones. Posteriormente, se implementa un caso práctico utilizando el algoritmo K-Nearest Neighbors (KNN) sobre un conjunto de datos clínicos simplificado.

1. Métricas de Clasificación

1.1 Accuracy (Exactitud)

Definición: Proporción de predicciones correctas sobre el total.

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

Interpretación: Indica qué tan frecuentemente el modelo acierta.

Ventajas: Intuitiva, fácil de calcular.

Limitaciones: No funciona bien con clases desbalanceadas.

1.2 Precision (Precisión)

Definición: Proporción de predicciones positivas que son correctas.

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

Interpretación: Mide la confiabilidad de las predicciones positivas.

Ventajas: Útil cuando el costo de los falsos positivos es alto.

Limitaciones: Ignora los falsos negativos.

1.3 Recall (Sensibilidad)

Definición: Proporción de positivos reales que se identifican correctamente.

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

Interpretación: Evalúa la capacidad de detectar todos los casos positivos.

Ventajas: Útil cuando importa minimizar falsos negativos.

Limitaciones: Puede aumentar a costa de reducir precision.

1.4 F1-score

Definición: Media armónica entre precision y recall.

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

Interpretación: Balancea precision y recall.

Ventajas: Útil con clases desbalanceadas.

Limitaciones: No refleja el desempeño en verdaderos negativos.

1.5 ROC-AUC

Definición: Área bajo la curva ROC, que relaciona TPR y FPR.

Interpretación: Mide la capacidad del modelo de diferenciar clases.

Ventajas: Robusta ante desbalance de clases.

Limitaciones: Menos intuitiva; requiere probabilidades.

2. Métricas de Regresión

2.1 MAE (Error Absoluto Medio)

Definición: Promedio del valor absoluto de los errores.

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

Interpretación: Indica el error promedio en la misma escala de la variable.

Ventajas: Fácil de interpretar; robusta a outliers.

Limitaciones: No penaliza fuertemente grandes errores.

2.2 RMSE (Raíz del Error Cuadrático Medio)

Definición: Raíz cuadrada del promedio del error cuadrático.

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

Interpretación: Penaliza fuertemente errores grandes.

Ventajas: Útil cuando grandes errores son críticos.

Limitaciones: Sensible a outliers.

Sección 2 – Solución de Caso con KNN

a. Preparación de Datos

Se toma el conjunto de datos proporcionado (glucosa, edad, etiqueta). Se divide en 70% entrenamiento y 30% prueba, y se normalizan las variables.

b. Implementación (Python + scikit-learn)

```
import pandas as pd
from sklearn.metrics import (
    accuracy_score, precision_score, recall_score,
    f1_score, roc_curve, auc, confusion_matrix
)
import matplotlib.pyplot as plt

# 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]
]

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

```
# División\ nX_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)

# Escalado
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Probar diferentes k
results = {}
for k in [1,3,5]:
    model = KNeighborsClassifier(n_neighbors=k)
    model.fit(X_train_scaled, y_train)
    y_pred = model.predict(X_test_scaled)
    results[k] = f1_score(y_test, y_pred)

best_k = max(results, key=results.get)
best_k

# Entrenar modelo final
model = KNeighborsClassifier(n_neighbors=best_k)
model.fit(X_train_scaled, y_train)
y_pred = model.predict(X_test_scaled)
y_prob = model.predict_proba(X_test_scaled)[:,1]

# Métricas
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
cm = confusion_matrix(y_test, y_pred)

# Curva ROC
fpr, tpr, _ = roc_curve(y_test, y_prob)
```

`roc_auc = auc(fpr, tpr)`

c. Evaluación

- Se calculan todas las métricas investigadas (accuracy, precision, recall, F1-score y AUC).
- Se genera matriz de confusión.
- Se genera curva ROC.

d. Análisis de resultados

- Generalmente, **k = 3** ofrece un equilibrio adecuado entre ruido y sesgo.
- Valores muy pequeños de k generan sobreajuste.
- Valores muy grandes suavizan demasiado la frontera de decisión.
- Posibles mejoras:
 - Incorporar más características clínicas.
 - Ajustar pesos según distancia.
 - Probar algoritmos alternativos como SVM o Random Forest.

Conclusiones

Las métricas de evaluación permiten entender de manera completa el desempeño de los modelos, cada una destacando distintos aspectos. En el caso práctico, el uso de KNN demostró cómo la selección adecuada de k y el preprocesamiento influyen directamente en el rendimiento. La experimentación evidenció que F1-score es una métrica robusta para determinar el mejor modelo en problemas con clases potencialmente desbalanceadas.

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.