Universidad Tecnológica de Chihuahua

EXTRACCIÓN DE CONOCIMIENTO EN BASES DE DATOS

III.2. Reporte de Métricas de Evaluación

Enrique Mascote
RICARDO ALONSO RIOS MONRREAL

Introducción

La evaluación del rendimiento es una etapa crítica en el ciclo de vida de cualquier proyecto de aprendizaje automático. No basta con entrenar un modelo; es necesario cuantificar qué tan bien generaliza ante datos nuevos. En este reporte, se exploran las métricas fundamentales para tareas de clasificación y regresión, analizando sus fórmulas y aplicaciones prácticas. Posteriormente, se aplican estos conceptos en un caso práctico utilizando el algoritmo *K-Nearest Neighbors* (KNN) sobre un conjunto de datos médicos para predecir una etiqueta binaria basada en niveles de glucosa y edad, buscando optimizar el modelo mediante el análisis del F1-score.

Investigación de métricas

Métricas de Clasificación

a. Accuracy (Exactitud)

- Definición: Es la proporción de predicciones correctas (tanto positivas como negativas) sobre el total de casos.
- Interpretación: Indica el porcentaje global de aciertos del modelo.
- Ventajas/Limitaciones: Es muy intuitiva, pero puede ser engañosa en datasets desbalanceados (ej. si el 90% de los datos son clase 0, el modelo puede tener 90% de accuracy prediciendo siempre 0, sin aprender nada).

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

b. Precision (Precisión)

- Definición: De todas las instancias que el modelo clasificó como positivas, ¿cuántas lo eran realmente?
- Interpretación: Mide la "calidad" de las predicciones positivas. Un valor bajo indica muchos falsos positivos.
- Ventajas/Limitaciones: Útil cuando el costo de un Falso Positivo es alto (ej. clasificar un correo legítimo como spam). No considera los falsos negativos.

$$\text{Precision} = \frac{\text{correctly classified actual positives}}{\text{everything classified as positive}} = \frac{TP}{TP + FP}$$

c. Recall (Sensibilidad)

- Definición: De todas las instancias que realmente eran positivas, ¿cuántas detectó el modelo?
- Interpretación: Mide la capacidad del modelo para encontrar todos los casos relevantes.
- Ventajas/Limitaciones: Crucial en medicina (ej. detectar enfermos), donde omitir un caso positivo es peligroso. Puede generar muchos falsos positivos si no se balancea con la precisión.

$$\text{Recall (or TPR)} = \frac{\text{correctly classified actual positives}}{\text{all actual positives}} = \frac{TP}{TP + FN}$$

d. F1-Score

- Definición: Es la media armónica entre Precision y Recall.
- Interpretación: Proporciona una métrica única que balancea ambas cualidades.
- Ventajas/Limitaciones: Es excelente para comparar clasificadores, especialmente en clases desbalanceadas, ya que penaliza si una de las dos métricas es muy baja.

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

Métricas de Regresión

a. MAE (Mean Absolute Error)

- Definición: El promedio de las diferencias absolutas entre la predicción y el valor real.
- Interpretación: Indica cuánto nos equivocamos en promedio en las mismas unidades de la variable original.

$$MAE = \frac{\sum_{i=1}^{n} \left| y_i - x_i \right|}{n}$$

- Ventajas/Limitaciones: Es robusto ante valores atípicos (outliers), pero no penaliza fuertemente los errores grandes.

b. RMSE (Root Mean Squared Error)

- Definición: La raíz cuadrada del promedio de los errores al cuadrado.
- Interpretación: Similar al MAE pero da más peso a los errores grandes.
- Ventajas/Limitaciones: Es útil si queremos evitar errores grandes a toda costa, pero es más sensible a outliers.

$$MSE = \frac{\sum_{i=1}^{n} (y_i - x_i)^2}{n}$$

Solución con KNN

a. Preparación de datos Se cargó la matriz de datos Matriz.csv. Se seleccionaron las columnas glucosa y edad como variables predictoras (features) y etiqueta como la variable objetivo (target).

- **División:** Se utilizó train_test_split de Scikit-Learn para separar el 70% de los datos para entrenamiento y el 30% restante para validación.
- **Escalamiento:** Dado que KNN es un algoritmo basado en distancias, es sensible a las magnitudes de las variables. Se aplicó StandardScaler para normalizar glucosa y edad (media 0, desviación estándar 1), evitando que la variable con valores más grandes domine el cálculo de la distancia.

b. Implementación Se entrenaron modelos KNN variando el hiperparámetro (número de vecinos) con los valores [3, 5, 7, 9]. El criterio de selección para el mejor modelo fue el **F1-score**, dado que buscamos un balance entre precisión y exhaustividad.

4. Resultados

Tras ejecutar las pruebas, se obtuvieron los siguientes rendimientos en el conjunto de prueba:

- **k=3:** F1-score = 0.8571
- **k=5:** F1-score = 0.8571
- **k=7:** F1-score = 1.0000
- **k=9:** F1-score = 1.0000

Se seleccionó como el mejor modelo (por ser el valor más bajo que alcanza el máximo rendimiento).

Métricas finales del modelo seleccionado (k = 7):

Métrica	Valor
Accuracy	1.00 (100%)
Precision	1.00 (100%)
Recall	1.00 (100%)
F1-Score	1.00 (100%)
AUC	1.00

Matriz de Confusión:

[[6 0]

 [0 3]]

Interpretación*:* El modelo clasificó correctamente los 6 casos negativos (Verdaderos Negativos) y los 3 casos positivos (Verdaderos Positivos) del set de prueba, sin cometer ningún error.

Curva ROC: Al obtener un AUC de 1.0, la curva ROC forma un ángulo perfecto en la esquina superior izquierda (tasa de verdaderos positivos = 1 y tasa de falsos positivos = 0), indicando una separación ideal entre las clases para este conjunto de datos.

Conclusiones y recomendaciones

Los resultados obtenidos muestran un rendimiento perfecto (F1-score de 1.0). Si bien esto es ideal teóricamente, en la práctica sugiere dos posibilidades:

1. El conjunto de datos es demasiado pequeño (solo 9 muestras en el set de prueba), lo que hace que la evaluación sea poco representativa estadísticamente.
2. Las clases son linealmente separables de manera muy clara en el espacio de glucosa y edad.

Recomendaciones:

- Aumentar el dataset: Para validar si el modelo es realmente robusto, se recomienda conseguir más registros históricos. Un test con solo 9 datos no garantiza que el modelo funcione igual de bien en producción.

- Probar Cross-Validation: En lugar de una sola división 70/30, usar validación cruzada (k-fold) daría una estimación más realista del error.

Script de Implementación

```python
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 accuracy_score, precision_score, recall_score, f1_score, roc_auc_score, confusion_matrix

# 1. Carga de datos
df = pd.read_csv('Matriz.csv')
X = df[['glucosa', 'edad']]
y = df['etiqueta']

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

# 3. Escalamiento (Normalización)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# 4. Búsqueda del mejor k
k_values = [3, 5, 7, 9]
best_k = 0
best_f1 = 0
best_model = None

print("Evaluando valores de k...")
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)
    print(f"k={k}: F1-score={f1:.4f}")

    if f1 > best_f1:
        best_f1 = f1
        best_k = k
        best_model = knn

# 5. Evaluación final
print(f"\nMejor k seleccionado: {best_k}")
y_pred_final = best_model.predict(X_test_scaled)
y_prob_final = best_model.predict_proba(X_test_scaled)[:, 1]

print("\n--- Métricas Finales ---")
print(f"Accuracy: {accuracy_score(y_test, y_pred_final):.4f}")
print(f"Precision: {precision_score(y_test, y_pred_final):.4f}")
print(f"Recall: {recall_score(y_test, y_pred_final):.4f}")
print(f"F1-Score: {f1_score(y_test, y_pred_final):.4f}")
print(f"AUC: {roc_auc_score(y_test, y_prob_final):.4f}")
```

Referencias

Clasificación: Exactitud, recuperación, precisión y métricas relacionadas. (n.d.). Google for Developers. https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall?hl=es-419

Madrigal, E. (2022, November 3). Conoce las métricas de precisión más comunes para Modelos de Regresión. *Grow Up*. https://www.growupcr.com/post/metricas-precision