



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO



Ejercicio de Lab5: Medidas de desempeño en ML

Alumno: Delgado Acosta Luis Bernardo

Franco Calderas Sergio Alberto

Grupo: 5BV1

Medidas de desempeño

Las medidas de desempeño son fundamentales para evaluar qué tan bien un modelo está funcionando al realizar predicciones. Dependiendo del problema (clasificación, regresión, etc.), estas métricas ayudan a entender si el modelo está tomando decisiones correctas.

La matriz de confusión: Es una tabla que muestra la cantidad de predicciones correctas e incorrectas del modelo, clasificado en:

- TP (True Positives): Casos donde el modelo predice verdadero y la etiqueta real también es verdadero.
- FP (False Positives): Casos donde el modelo predice falso pero la etiqueta real es verdadero.
- FN (False Negatives): Casos donde el modelo predice falso pero la etiqueta real es falso.
- TN (True Negatives): Casos donde el modelo predice verdadero y la etiqueta real es falso.

Para el desarrollo de la práctica se programarán las funciones para obtener la matriz de confusión de un dataset con dos clases y a partir de la matriz de confusión calcular las siguientes métricas:

- Accuracy: Mide el porcentaje de predicciones correctas
- Error: Indica el porcentaje de predicciones incorrectas.
- El Positive Predictive Value (precision), también conocido como precisión, mide la proporción de predicciones positivas que son correctas
- True Positive Rate (recall): Mide la proporción de casos positivos reales que el modelo ha identificado correctamente como positivos.
- True Negative Rate: Mide la proporción de negativos correctamente identificados.
- False Positive Rate: Mide la proporción de negativos mal clasificados como positivos.
- False Negative Rate: Mide la proporción de positivos mal clasificados como negativos.
- F1-Score: Es una métrica que combina precision y recall en una sola medida balanceada.

A continuación, se muestran las funciones para calcular la matriz de confusión y las medidas de desempeño.

```
# Funcion para calcular la matriz de confusion
def matriz_confusion(y_true, y_pred):
    TP = FN = FP = TN = 0
    for i in range(len(y_true)):
        if y_true[i] == 1 and y_pred[i] == 1:
            TP += 1
        elif y_true[i] == 1 and y_pred[i] == 0:
            FN += 1
        elif y_true[i] == 0 and y_pred[i] == 1:
            FP += 1
        elif y_true[i] == 0 and y_pred[i] == 0:
            TN += 1
    return TP, FN, FP, TN
```

Fig.1 Función para calcular la matriz de confusión

```
# Funcion para calcular las metricas
def calcular_metricas(TP, TN, FP, FN):
    precision = TP / (TP + FP) if (TP + FP) > 0 else 0
    recall = TP / (TP + FN) if (TP + FN) > 0 else 0
    positive_predictive_value = precision
    true_positive_rate = recall
    true_negative_rate = TN / (TN + FP) if (TN + FP) > 0 else 0
    false_positive_rate = FP / (FP + TN) if (FP + TN) > 0 else 0
    false_negative_rate = FN / (FN + TP) if (FN + TP) > 0 else 0
    f1_score = (2 * precision * recall) / (precision + recall) if (precision + recall) > 0 else 0

    # Calcular accuracy y error rate
    total = TP + TN + FP + FN
    accuracy = (TP + TN) / total if total > 0 else 0
    error_rate = (FP + FN) / total if total > 0 else 0

    return {
        "Accuracy": accuracy,
        "Error Rate": error_rate,
        "Precision": precision,
        "Recall": recall,
        "Positive Predictive Value": positive_predictive_value,
        "True Positive Rate": true_positive_rate,
        "True Negative Rate": true_negative_rate,
        "False Positive Rate": false_positive_rate,
        "False Negative Rate": false_negative_rate,
        "F1-Score": f1_score,
    }
```

Fig.2 Función para calcular las métricas

Enlaces a los repositorios

Bernardo: <https://github.com/BernardoD07/Machine-Learning-ESCOM.git>

Sergio: <https://github.com/SergioCalderas/Trabajos-Machine-Learning.git>

PARTE 2:

Se encontró la librería '**sklearn-metrics**' para calcular las métricas de la parte 1. A continuación se ejemplifica con un arreglo de clases reales y otro de predicciones.

```
y_true = [1, 0, 1, 1, 0, 1, 0, 0]
y_pred = [1, 0, 0, 0, 0, 1, 0, 1]
```

Fig.3 Ejemplificación

Se utiliza la función '**confusion_matrix (y_true, y_pred)**' para sacar la matriz de confusión

```
Matriz de confusión:
[[3 1]
 [2 2]]
```

Fig.4 Matriz de confusión

Se observa que 3 fueron clasificados como verdadero positivo, 2 verdadero negativo, y 1 de falso positivo, y 2 falso negativo. Tenemos las siguientes funciones para las demás métricas:

- **accuracy_score()**: Mide el porcentaje de predicciones correctas sobre el total de casos evaluados. Es útil cuando las clases están balanceadas.
- **precision_score()**: Mide la proporción de predicciones positivas que fueron correctas. Es crítica cuando es más importante minimizar los falsos positivos
- **recall_score()**: Evalúa qué proporción de los casos positivos reales fueron correctamente identificados. Es esencial en problemas donde perder casos positivos es costoso, por ejemplo, en diagnósticos médicos.
- **f1_score()**: Es la media armónica entre la precisión y recall, proporcionando un equilibrio entre ambas métricas. Es especialmente útil cuando las clases están desbalanceadas, ya que no favorece ninguna métrica sobre la otra.

```
Accuracy: 0.62
Error:0.38
Presicion: 0.67
Recall: 0.50
F1 Score: 0.57
```

Fig.5 Resultados de métricas