# SESIÓN MÉTRICAS DE DESEMPEÑO DE UN ALGORITMO

#### **CONTENIDOS:**

- 1. Métricas de desempeño de un algoritmo de regresión:
  - o MAE (Error Absoluto Medio).
  - MSE (Error Cuadrático Medio).
  - o RMSE (Raíz del Error Cuadrático Medio).
  - o R<sup>2</sup> (Coeficiente de Determinación).
- 2. Métricas de desempeño en un algoritmo de clasificación:
  - o Matriz de confusión.
  - Precisión y exactitud.
  - o Sensibilidad y especificidad.
  - Curva ROC-AUC.

## MÉTRICAS DE DESEMPEÑO DE UN ALGORITMO DE REGRESIÓN

## MAE (Error Absoluto Medio)

El **Error Absoluto Medio (MAE)** es una de las métricas más comunes utilizadas para evaluar el desempeño de modelos de regresión. Mide el promedio de los errores absolutos entre los valores reales  $(y_i)$  y los valores predichos  $(\tilde{y}_i)$  por el modelo. A diferencia del **Error Cuadrático Medio (MSE)**, el MAE no eleva al cuadrado los errores, lo que lo hace menos sensible a valores atípicos (outliers).

#### Fórmula del MAE

La fórmula del MAE es la siguiente:

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

Ilustración 1 Fórmula MAE.

- y<sub>i</sub>: Valor real de la observación i.
- $\hat{y}_i$ : Valor predicho por el modelo para la observación i.
- n: Número total de observaciones.

## Interpretación del MAE

- Un MAE más bajo indica que el modelo tiene un mejor ajuste, ya que los errores absolutos entre las predicciones y los valores reales son menores.
- Un MAE más alto sugiere que el modelo tiene un peor ajuste, con errores más grandes en las predicciones.
- **Unidad**: El MAE tiene la misma unidad que la variable de salida, lo que facilita su interpretación en el contexto del problema.

## Ventajas del MAE

- 1. **Menos sensible a valores atípicos**: Al no elevar los errores al cuadrado, el MAE no se ve tan afectado por valores extremos en los datos.
- 2. **Fácil de interpretar**: Al ser un promedio de errores absolutos, es más intuitivo que otras métricas como el MSE o RMSE.
- No penaliza en exceso los errores grandes: A diferencia del MSE, el MAE no castiga tanto los errores grandes, lo que puede ser útil en ciertos contextos donde los valores atípicos no son significativos.

## Desventajas del MAE

- No diferencia entre errores grandes y pequeños: Al no elevar los errores al cuadrado, el MAE trata todos los errores por igual, lo que puede ser un problema en aplicaciones donde los errores grandes son más críticos.
- 2. **No es diferenciable**: El valor absoluto no es una función diferenciable en todos los puntos, lo que puede complicar su uso en algoritmos de optimización que requieren derivadas.

## Ejemplo de uso del MAE

Supongamos que tenemos un modelo que predice el precio de casas. Los valores reales y predichos son los siguientes:

Observación	Valor Real (y <sub>i</sub> )	Valor Predicho (ŷ <sub>i</sub> )	Error Absoluto $( y_i - \hat{y_i} )$
1	200,000	195,000	5,000
2	300,000	310,000	10,000
3	250,000	245,000	5,000
4	400,000	390,000	10,000

El MAE se calcula como:

$$MAE = \frac{5,000 + 10,000 + 5,000 + 10,000}{4} = \frac{30,000}{4} = 7,500$$

Esto significa que, en promedio, el modelo se desvía 7,500 unidades (en este caso, dólares) de los valores reales.

## MSE (Error Cuadrático Medio)

El **Error Cuadrático Medio (MSE)** es otra métrica ampliamente utilizada para evaluar modelos de regresión. A diferencia del MAE, el MSE eleva al cuadrado los errores entre los valores reales y los predichos, lo que penaliza más los errores grandes. Esto hace que el MSE sea más sensible a valores atípicos, ya que los errores grandes tienen un impacto desproporcionado en el valor final de la métrica.

#### Fórmula del MSE

La fórmula del MSE es la siguiente:

$$ext{MSE} = rac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Ilustración 3 Fórmula MSE.

- y<sub>i</sub>: Valor real de la observación i.
- $\hat{y}_i$ : Valor predicho por el modelo para la observación *i*.
- n: Número total de observaciones.

## Interpretación del MSE

- **Un MSE más bajo** indica que el modelo tiene un mejor ajuste, ya que los errores cuadráticos entre las predicciones y los valores reales son menores.
- Un MSE más alto sugiere que el modelo tiene un peor ajuste, con errores más grandes en las predicciones.
- **Unidad**: El MSE tiene unidades cuadradas de la variable de salida, lo que puede dificultar su interpretación en el contexto del problema.

## Ventajas del MSE

- Penaliza los errores grandes: Al elevar los errores al cuadrado, el MSE castiga más los errores grandes, lo que puede ser útil en aplicaciones donde los errores grandes son más críticos.
- 2. **Diferenciable**: El MSE es una función diferenciable, lo que facilita su uso en algoritmos de optimización que requieren derivadas, como el descenso de gradiente.

3. **Ampliamente utilizado**: Es una métrica estándar en muchos problemas de regresión, lo que facilita la comparación entre modelos.

## Desventajas del MSE

- 1. **Sensible a valores atípicos**: Debido a que eleva los errores al cuadrado, el MSE es muy sensible a valores atípicos, lo que puede distorsionar la evaluación del modelo.
- Unidades cuadradas: Al tener unidades cuadradas, el MSE puede ser difícil de interpretar en el contexto del problema, especialmente cuando se compara con otras métricas como el MAE.

## Ejemplo de uso del MSE

Utilizando el mismo ejemplo anterior de predicción de precios de casas:

Observación	Valor Real (y <sub>i</sub> )	Valor Predicho (ŷ <sub>i</sub> )	Error Cuadrático (y <sub>i</sub> − ŷ <sub>i</sub> )²
1	200,000	195,000	25,000,000
2	300,000	310,000	100,000,000
3	250,000	245,000	25,000,000
4	400,000	390,000	100,000,000

El MSE se calcula como:

$$MSE = \frac{25,000,000 + 100,000,000 + 25,000,000 + 100,000,000}{4} = \frac{250,000,000}{4} = 62,500,000$$

Esto significa que, en promedio, el modelo tiene un error cuadrático de 62,500,000 unidades cuadradas (en este caso, dólares al cuadrado).

## RMSE (Raíz del Error Cuadrático Medio)

El Raíz del Error Cuadrático Medio (RMSE) es una métrica ampliamente utilizada para evaluar el desempeño de modelos de regresión. Es la raíz cuadrada del Error Cuadrático Medio (MSE), lo que la convierte en una medida de error que tiene las mismas unidades que la variable de salida. Esto facilita su interpretación en el contexto del problema.

#### Fórmula del RMSE

La fórmula del RMSE es la siguiente:

$$ext{RMSE} = \sqrt{rac{1}{n}\sum_{i=1}^n(y_i - \hat{y}_i)^2}$$

Ilustración 5 Fórmula RMSE.

- y<sub>i</sub>: Valor real de la observación i.
- $\hat{y}_i$ : Valor predicho por el modelo para la observación i.
- n: Número total de observaciones.

## Interpretación del RMSE

- Un RMSE más bajo indica que el modelo tiene un mejor ajuste, ya que los errores entre las predicciones y los valores reales son menores.
- Un RMSE más alto sugiere que el modelo tiene un peor ajuste, con errores más grandes en las predicciones.
- **Unidad**: El RMSE tiene las mismas unidades que la variable de salida, lo que facilita su interpretación en el contexto del problema.

## Ventajas del RMSE

1. **Misma unidad que la variable de salida**: Al ser la raíz cuadrada del MSE, el RMSE tiene las mismas unidades que la variable de salida, lo que lo hace más interpretable que el MSE.

- 2. **Penaliza los errores grandes**: Al igual que el MSE, el RMSE penaliza más los errores grandes, lo que puede ser útil en aplicaciones donde los errores grandes son más críticos.
- 3. **Ampliamente utilizado**: Es una métrica estándar en muchos problemas de regresión, lo que facilita la comparación entre modelos.

## Desventajas del RMSE

- 1. **Sensible a valores atípicos**: Debido a que se basa en el MSE, el RMSE es sensible a valores atípicos, lo que puede distorsionar la evaluación del modelo.
- 2. **No diferencia entre errores grandes y pequeños**: Al igual que el MSE, el RMSE trata los errores grandes de manera más severa, lo que puede no ser deseable en algunos contextos.

### Ejemplo de uso del RMSE

Supongamos que tenemos un modelo que predice el precio de casas. Los valores reales y predichos son los siguientes:

Observación	Valor Real (y <sub>i</sub> )	Valor Predicho $(\hat{y_i})$	Error Cuadrático (y <sub>i</sub> − ŷ <sub>i</sub> )²
1	200,000	195,000	25,000,000
2	300,000	310,000	100,000,000
3	250,000	245,000	25,000,000
4	400,000	390,000	100,000,000

El RMSE se calcula como:

$$RMSE = \sqrt{\frac{25,000,000 + 100,000,000 + 25,000,000 + 100,000,000}{4}} = \sqrt{\frac{250,000,000}{4}} = \sqrt{\frac{62,500,000}{62,500,000}} \approx 7,905.5$$

Esto significa que, en promedio, el modelo tiene un error de **7,905.5 dólares** en sus predicciones.

## R<sup>2</sup> (Coeficiente de Determinación)

El **Coeficiente de Determinación (R²)** es una métrica que mide la proporción de la varianza en la variable dependiente que es explicada por el modelo. Varía entre 0 y 1, donde un valor cercano a 1 indica que el modelo explica una gran parte de la varianza, mientras que un valor cercano a 0 indica un mal ajuste.

#### Fórmula del R<sup>2</sup>

La fórmula del R<sup>2</sup> es la siguiente:

$$R^2 = 1 - rac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - ar{y})^2}$$

Ilustración 7 Fórmula R<sup>2</sup>.

- y<sub>i</sub>: Valor real de la observación i.
- $\hat{y}_i$ : Valor predicho por el modelo para la observación i.
- **Y**: Media de los valores reales.
- n: Número total de observaciones.

## Interpretación del R<sup>2</sup>

- Un valor de R<sup>2</sup> cercano a 1 indica que el modelo explica una gran parte de la varianza en la variable dependiente.
- Un valor de R<sup>2</sup> cercano a 0 indica que el modelo no explica bien la varianza y tiene un mal ajuste.
- Un valor negativo de  $R^2$  es posible en casos donde el modelo es peor que un modelo simple que siempre predice la media.

## Ventajas del R<sup>2</sup>

- 1. Interpretación intuitiva: El R2 es fácil de interpretar, ya que su valor oscila entre 0 y 1.
- Comparación entre modelos: Permite comparar la capacidad explicativa de diferentes modelos.
- 3. **No depende de la escala**: El R<sup>2</sup> es una métrica adimensional, lo que facilita su uso en diferentes contextos.

#### Desventajas del R<sup>2</sup>

- 1. **No indica causalidad**: Un alto valor de R<sup>2</sup> no implica que el modelo sea causalmente correcto.
- 2. **Sensible a la complejidad del modelo**: El R² puede aumentar simplemente añadiendo más variables al modelo, incluso si no son relevantes.
- 3. **No detecta sobreajuste**: Un modelo sobreajustado puede tener un R² alto en los datos de entrenamiento, pero un rendimiento pobre en datos no vistos.

## Ejemplo de uso del R<sup>2</sup>

Supongamos que tenemos un modelo que predice el precio de casas. Los valores reales y predichos son los siguientes:

Observación	Valor Real (y <sub>i</sub> )	Valor Predicho $(\hat{y_i})$	Error Cuadrático (y₁− ŷ₁)²
1	200,000	195,000	25,000,000
2	300,000	310,000	100,000,000
3	250,000	245,000	25,000,000
4	400,000	390,000	100,000,000

La media de los valores reales  $(\bar{Y})$  es:

$$ar{y} = rac{200,000 + 300,000 + 250,000 + 400,000}{4} = 287,500$$

Ilustración 9 media valores reales.

El R<sup>2</sup> se calcula como:

$$R^2 = 1 - \frac{25,000,000 + 100,000,000 + 25,000,000 + 100,000,000}{(200,000 - 287,500)^2 + (300,000 - 287,500)^2 + (250,000 - 287,500)^2 + (400,000 - 287,500)^2}$$

Ilustración 10 ejemplo R<sup>2</sup> (1)

$$R^2 = 1 - rac{250,000,000}{7,656,250,000+156,250,000+1,406,250,000+12,656,250,000} = 1 - rac{250,000,000}{21,875,000,000} pprox 0.988$$

Ilustración 11 ejemplo R<sup>2</sup> (2)

Esto significa que el modelo explica aproximadamente el 98.8% de la varianza en los precios de las casas.

## Implementación en Python

Veamos cómo implementar el cálculo del MAE, MSE, RMSE y el R<sup>2</sup> en Python utilizando la librería Scikit-learn:

```
# Importar librerías necesarias
from sklearn.metrics import mean_squared_error, mean_absolute_error,
r2_score
import numpy as np

# Valores reales y predichos
y_true = np.array([3, 5, 7, 9, 11])
y_pred = np.array([2.8, 5.1, 7.2, 8.9, 10.5])

# Calcular MSE
```

```
mse = mean_squared_error(y_true, y_pred)
print(f"MSE: {mse}")

# Calcular MAE
mae = mean_absolute_error(y_true, y_pred)
print(f"MAE: {mae}")

# Calcular RMSE
rmse = np.sqrt(mean_squared_error(y_true, y_pred))
print(f"RMSE: {rmse}")

# Calcular R²
r2 = r2_score(y_true, y_pred)
print(f"R²: {r2}")
```

## Salida Esperada:

R2: 0.99125

## MÉTRICAS DE DESEMPEÑO EN UN ALGORITMO DE CLASIFICACIÓN

En problemas de clasificación, es fundamental evaluar el rendimiento del modelo utilizando métricas que nos permitan entender cómo se están clasificando correctamente (o incorrectamente) las instancias. A continuación, veremos dos de las métricas más importantes: la **Matriz de Confusión** y las métricas derivadas de ella, como la **Precisión** y la **Exactitud**.

#### Matriz de Confusión

La **Matriz de Confusión** es una tabla que resume el desempeño de un modelo de clasificación al comparar las predicciones con los valores reales. Es especialmente útil para problemas de clasificación binaria, pero puede extenderse a problemas multiclase.

#### Estructura de la Matriz de Confusión

Para un problema de clasificación binaria, la matriz de confusión tiene la siguiente estructura:

	Predicción Negativa (0)	Predicción Positiva (1)
Real Negativo (0)	Verdaderos Negativos (VN)	Falsos Positivos (FP)
Real Positivo (1)	Falsos Negativos (FN)	Verdaderos Positivos (VP)

- Verdaderos Positivos (VP): Casos en los que el modelo predijo correctamente la clase positiva.
- Falsos Positivos (FP): Casos en los que el modelo predijo incorrectamente la clase positiva (errores de tipo I).
- Falsos Negativos (FN): Casos en los que el modelo predijo incorrectamente la clase negativa (errores de tipo II).
- Verdaderos Negativos (VN): Casos en los que el modelo predijo correctamente la clase negativa.

## Ejemplo de Matriz de Confusión

Supongamos que tenemos un modelo que predice si un correo es spam (clase positiva) o no spam (clase negativa). La matriz de confusión podría ser:

	Predicción No Spam (0)	Predicción Spam(0)
Real No Spam (0)	50 (VN)	10 (FP)
Real Spam (0)	5 (FN)	100 (VP)

- VN (Verdaderos Negativos): 50 correos no spam fueron correctamente clasificados como no spam.
- FN (Falsos Negativos): 5 correos spam fueron incorrectamente clasificados como no spam.
- FP (Falsos Positivos): 10 correos no spam fueron incorrectamente clasificados como spam.
- VP (Verdaderos Positivos): 100 correos spam fueron correctamente clasificados como spam.

## Precisión y Exactitud

A partir de la matriz de confusión, se pueden calcular métricas como la **Precisión** y la **Exactitud**, que nos ayudan a entender mejor el rendimiento del modelo.

## Precisión (Precision)

La Precisión mide la proporción de predicciones positivas que son correctas. Es útil cuando los falsos positivos (FP) son costosos o no deseables.

#### Fórmula:

$$ext{Precisión} = rac{ ext{VP}}{ ext{VP} + ext{FP}}$$

Ilustración 12 Fórmula Precisión.

## • Interpretación:

- Un valor alto de precisión indica que el modelo tiene pocos falsos positivos.
- Es especialmente importante en aplicaciones como el diagnóstico médico, donde los falsos positivos pueden tener consecuencias graves.

## • Ejemplo:

Usando la matriz de confusión anterior:

$$\text{Precisión} = \frac{50}{50+5} = \frac{50}{55} \approx 0.909$$

Ilustración 13 ejemplo Precisión.

Esto significa que el 90.9% de las predicciones de spam fueron correctas.

## **Exactitud (Accuracy)**

La Exactitud mide la proporción de predicciones correctas (tanto positivas como negativas) sobre el total de predicciones. Es una métrica general que nos dice qué tan bien se está desempeñando el modelo en términos globales.

#### Fórmula:

$$\mathrm{Exactitud} = \frac{\mathrm{VP} + \mathrm{VN}}{\mathrm{VP} + \mathrm{FP} + \mathrm{VN} + \mathrm{FN}}$$

Ilustración 14 Fórmula Exactitud.

## • Interpretación:

- Un valor alto de exactitud indica que el modelo está haciendo muchas predicciones correctas.
- Sin embargo, en conjuntos de datos desbalanceados, la exactitud puede ser engañosa, ya que no distingue entre errores en las clases mayoritarias y minoritarias.

#### Ejemplo:

Usando la matriz de confusión anterior:

Exactitud = 
$$\frac{50 + 100}{50 + 5 + 100 + 10} = \frac{150}{165} \approx 0.909$$

Ilustración 15 Ejemplo Exactitud.

Esto significa que el 90.9% de todas las predicciones fueron correctas.

## Sensibilidad y Especificidad

Estas dos métricas son fundamentales para evaluar el rendimiento de un modelo de clasificación, especialmente en problemas donde las clases están desbalanceadas o donde los costos de los errores son diferentes (por ejemplo, en diagnóstico médico).

#### 1. Sensibilidad (Recall o Tasa de Verdaderos Positivos)

La **Sensibilidad** mide la proporción de casos positivos que el modelo identificó correctamente. Es decir, cuántos de los casos que realmente son positivos fueron correctamente clasificados como positivos por el modelo.

#### Fórmula:

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

Ilustración 16 Fórmula Sensibilidad.

#### Donde:

- o **VP (Verdaderos Positivos)**: Casos positivos correctamente clasificados.
- FN (Falsos Negativos): Casos positivos incorrectamente clasificados como negativos.

## Interpretación:

- Un valor alto de sensibilidad indica que el modelo es bueno para identificar casos positivos.
- Es especialmente importante en aplicaciones donde los falsos negativos son críticos, como en el diagnóstico de enfermedades (donde no detectar un caso positivo puede tener consecuencias graves).

## Ejemplo:

Supongamos que en una matriz de confusión tenemos:

- o VP = 50
- o FN = 10

$$Sensibilidad = \frac{50}{50+10} = \frac{50}{60} \approx 0.833$$

Ilustración 17 ejemplo sensibilidad.

Esto significa que el modelo identificó correctamente el 83.3% de los casos positivos.

## 2. Especificidad (Tasa de Verdaderos Negativos)

La Especificidad mide la proporción de casos negativos que el modelo identificó correctamente. Es decir, cuántos de los casos que realmente son negativos fueron correctamente clasificados como negativos por el modelo.

#### Fórmula:

$$\operatorname{Especificidad} = \frac{\operatorname{VN}}{\operatorname{VN} + \operatorname{FP}}$$

Ilustración 18 Fórmula Especificidad.

#### Donde:

- VN (Verdaderos Negativos): Casos negativos correctamente clasificados.
- FP (Falsos Positivos): Casos negativos incorrectamente clasificados como positivos.

## Interpretación:

- Un valor alto de especificidad indica que el modelo es bueno para identificar casos negativos.
- Es especialmente importante en aplicaciones donde los falsos positivos son costosos, como en pruebas de seguridad (donde un falso positivo puede generar alertas innecesarias).

## • Ejemplo:

Supongamos que en una matriz de confusión tenemos:

- o VN = 100
- o FP = 5

$$ext{Especificidad} = rac{100}{100+5} = rac{100}{105} pprox 0.$$

Ilustración 19 Ejemplo Especificidad.

Esto significa que el modelo identificó correctamente el 95.2% de los casos negativos.

## **Curva ROC-AUC**

La Curva ROC (Receiver Operating Characteristic) y el AUC (Área Bajo la Curva) son herramientas gráficas y métricas que permiten evaluar el rendimiento de un modelo de clasificación en función de su sensibilidad y especificidad para diferentes umbrales de clasificación.

#### 1. Curva ROC

- ¿Qué es?
  - La curva ROC es una gráfica que muestra la relación entre la tasa de verdaderos positivos (Sensibilidad) y la tasa de falsos positivos (1 - Especificidad) para diferentes umbrales de clasificación.
  - El eje X representa la tasa de falsos positivos (FPR):

$$FPR = \frac{FP}{FP + VN}$$

Ilustración 20 Fórmula FPR.

El eje Y representa la tasa de verdaderos positivos (TPR o Sensibilidad):

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

Ilustración 21 Fórmula TPR o Sensibilidad.

## • Interpretación:

- Un modelo perfecto tendría una curva ROC que pasa por el punto (0, 1), es decir,
   100% de sensibilidad y 100% de especificidad.
- Un modelo aleatorio tendría una curva ROC que sigue la línea diagonal desde (0, 0) hasta (1, 1).

## Implementación en Python

A continuación, veremos cómo implementar y calcular la **matriz de** confusión, precisión, exactitud, sensibilidad, especificidad, curva ROC y AUC.

```
from sklearn.metrics import (
    confusion_matrix,
    precision_score,
    accuracy_score,
    recall_score,
    roc_curve,
```

```
roc_auc_score,
import matplotlib.pyplot as plt
import numpy as np
# Valores reales y predichos
y_true = np.array([0, 1, 1, 0, 1, 0, 1, 0, 1, 0])
y_pred = np.array([0, 1, 0, 0, 1, 0, 1, 0, 1, 0])
# Probabilidades predichas (para la curva ROC)
y_scores = np.array([0.1, 0.9, 0.4, 0.2, 0.7, 0.3, 0.6, 0.4, 0.8, 0.15])
# Matriz de Confusión
conf matrix = confusion matrix(y true, y pred)
print("Matriz de Confusión:")
print(conf_matrix)
# Precisión y Exactitud
precision = precision_score(y_true, y_pred)
accuracy = accuracy score(y true, y pred)
print(f"Precision: {precision}")
print(f"Exactitud: {accuracy}")
# Sensibilidad (Recall) y Especificidad
sensitivity = recall_score(y_true, y_pred) # Sensibilidad es lo mismo que
Recall
specificity = recall score(
    y_true, y_pred, pos_label=0
) # Especificidad es Recall para la clase negativa
print(f"Sensibilidad: {sensitivity}")
print(f"Especificidad: {specificity}")
# Curva ROC y AUC
fpr, tpr, thresholds = roc_curve(y_true, y_scores)
auc = roc_auc_score(y_true, y_scores)
print(f"AUC: {auc}")
# Graficar la curva ROC
plt.plot(fpr, tpr, label=f"Curva ROC (AUC = {auc:.2f})")
plt.plot([0, 1], [0, 1], linestyle="--", label="Clasificador aleatorio")
plt.xlabel("Tasa de Falsos Positivos (FPR)")
```

```
plt.ylabel("Tasa de Verdaderos Positivos (TPR)")
plt.title("Curva ROC")
plt.legend()
plt.show()
```

## Salida Esperada:

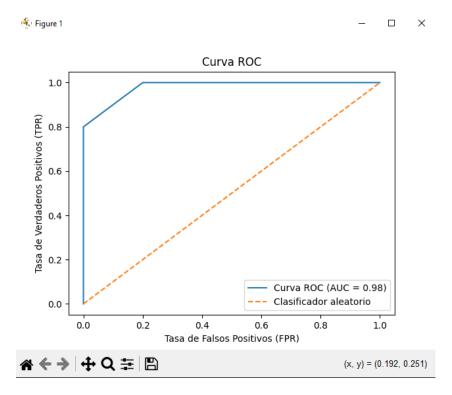
Terminal:

Matriz de Confusión:

[[5 0] [1 4]]

Precisión: 1.0 Exactitud: 0.9 Sensibilidad: 0.8 Especificidad: 1.0 AUC: 0.980000000000000000

Grafico:



## Explicación de los resultados

#### 1. Matriz de Confusión:

• [[5 0]

[1 4]]

## Primera fila:

- 5: Verdaderos Negativos (VN) → El modelo predijo correctamente 5 casos como negativos.
- 0: Falsos Positivos (FP) → El modelo no cometió errores al predecir la clase negativa.

## Segunda fila:

- 1: Falsos Negativos (FN) → El modelo predijo incorrectamente 1 caso positivo como negativo.
- 4: Verdaderos Positivos (VP) → El modelo predijo correctamente 4 casos como positivos.

## 2. Precisión:

 1.0: El modelo no tiene falsos positivos, por lo que todas las predicciones positivas son correctas.

## 3. Exactitud:

o **0.9**: El 90% de las predicciones fueron correctas.

## 4. Sensibilidad (Recall):

o **0.8**: El modelo identificó correctamente el 80% de los casos positivos.

## 5. Especificidad:

o **1.0**: El modelo identificó correctamente el 100% de los casos negativos.

## 6. **AUC**:

 0.98000000000001: El modelo tiene una excelente capacidad para distinguir entre las clases positiva y negativa.

## 7. Curva ROC:

- La línea diagonal representa el rendimiento de un clasificador aleatorio, donde el AUC sería 0.5.
- La curva ROC del modelo está muy por encima de esta línea, lo que confirma que el modelo es significativamente mejor que un clasificador aleatorio.