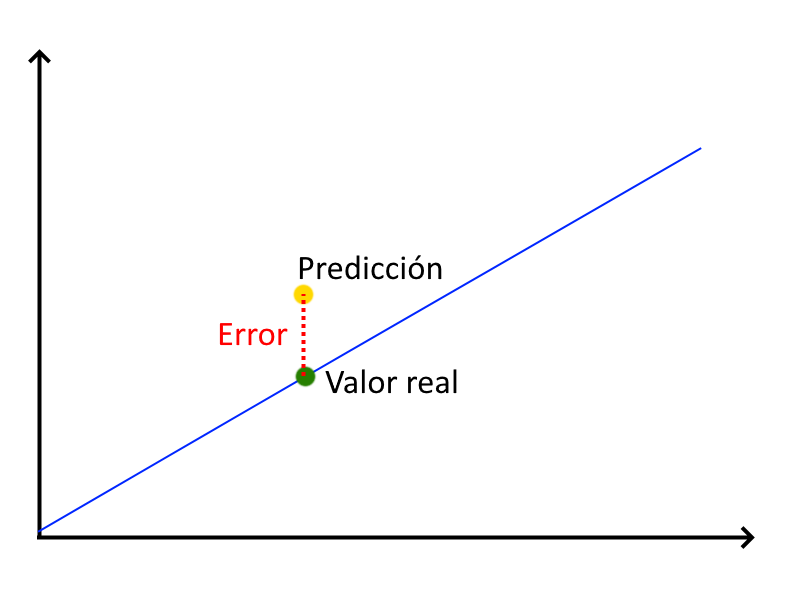
Métricas de error



Adrián Yared Armas de la Nuez

**Contenido**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

[**1. Enunciado 2**](#_2c35568x6wg8)

[**2 Explicación 2**](#_v91eomu50iy0)

[**2.1 Enunciado 2**](#_aeclegx8mmta)

[**2.1.1 Código 3**](#_b8ieovu82x9k)

[**2.1.2 Ejecución 3**](#_vvqds62za17u)

[**3 Explicación 3**](#_dzqv868o8k66)

[**3.1 Enunciado 4**](#_v1z5bjew5oeu)

[**3.1.1 Código 4**](#_59nyyjsoekm8)

[**2.1.2 Ejecución 4**](#_4b5bz2hgllo3)

[**4.1 Enunciado 5**](#_g5j8cri4z9z3)

[**4.1.1 Código 5**](#_oqbtda2ysnct)

[**4.1.2 Ejecución 6**](#_br6pjthqpyy6)

[**5.1 Enunciado 6**](#_eq9m9k8j7bl3)

[**5.1.1 Código 6**](#_nuopue62ig3i)

[**5.1.2 Ejecución 7**](#_mklc7ctqzi3s)

[**6 Explicación 7**](#_aimfoau1y94h)

[**6.1 Enunciado 7**](#_d0sx2mxosbum)

[**6.1.1 Código 7**](#_9pdoe8qjdumc)

[**6.1.2 Ejecución 8**](#_5icstzq4gn73)

[**7.1 Enunciado 8**](#_talrupeq24zn)

[**7.1.1 Código 8**](#_g1d2aomnanmv)

[**7.1.2 Ejecución 9**](#_ijjw5b5wzxqw)

[**8. Conclusión 9**](#_eh9gsbwevxwr)

[**9. Github y Colab 9**](#_lmdngf8dlin3)

## 

## 

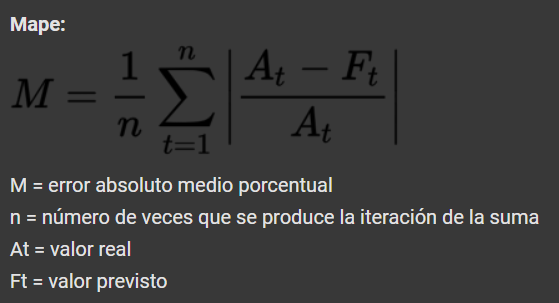
## **1. Enunciado**

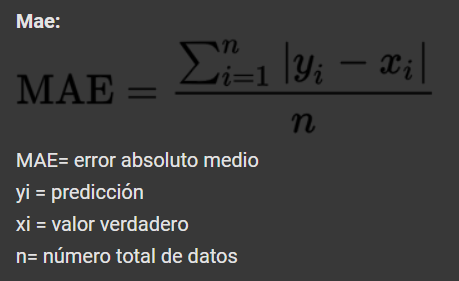
A partir de los ejemplos indicados en el siguiente cuaderno (Act\_3\_1) realizar los ejercicios

propuestos

Título: Act\_3\_1\_Metricas de error\_Sin soluciones.ipynb

Url:<https://colab.research.google.com/drive/1U6kfjRn4Lm8rRKzRL65KaHncZ5GQNgKH?usp=sharing>





## **2 Explicación**

#Error absoluto medio - mae

from sklearn.metrics import mean\_absolute\_error

# Ejemplo 1

y\_true = [2.5, 0, 2, 8]

y\_pred = [2.5, 0, 2, 8]

mean\_absolute\_error(y\_true, y\_pred)

–

Resultado : 0.0

### **2.1 Enunciado**

# Ejer 1: Crear dos vectores/arrays de valores ciertos y valores previstos, y calcular el mae.

# ¿Qué valores deberían de tener el array pred para que el error fuese 0? ¿y para que error fuese de 0.25?

#### **2.1.1 Código**

from sklearn.metrics import mean\_absolute\_error

# Ejemplo: Valores reales y previstos

y\_true = [3.0, -0.5, 2.0, 7.0]

y\_pred = [2.5, 0.0, 2.0, 8.0]

# Calcular el mae

mae = mean\_absolute\_error(y\_true, y\_pred)

print("MAE:", mae)

# Error 0 (los valores predichos deben ser iguales a los reales)

y\_pred\_zero\_error = [3.0, -0.5, 2.0, 7.0]

mae\_zero = mean\_absolute\_error(y\_true, y\_pred\_zero\_error)

print("MAE con error 0:", mae\_zero)

# Ejemplo para error 0.25: Ajustamos los valores predichos de forma que la media de los errores absolutos sea 0.25

y\_pred\_025\_error = [3.1, -0.5, 2.0, 7.0] # La diferencia total es 0.25 x n de valores (0.25 x 8 = 1)

mae\_025 = mean\_absolute\_error(y\_true, y\_pred\_025\_error)

print("MAE con error 0.25:", mae\_025)

#### **2.1.2 Ejecución**



## **3 Explicación**

# Ejemplo 2

y\_true = [[0.5, 1], [-1, 1], [7, -6]]

y\_pred = [[0, 2], [-1, 2], [8, -5]]

mean\_absolute\_error(y\_true, y\_pred)

–

Resultado : 0.75

### **3.1 Enunciado**

# Ejer 2: Crear dos matrices/arrays de valores ciertos y valores previstos, y calcular el mae.

# ¿Qué valores deberían de tener el array pred para que el error fuese 0? ¿y para que error fuese de 0.33?

#### **3.1.1 Código**

from sklearn.metrics import mean\_absolute\_error

# Matrices de ejemplo: Valores reales y previstos

y\_true = [[0.5, 1], [-1, 1], [7, -6]]

# Calcular el MAE original

y\_pred = [[0, 2], [-1, 2], [8, -5]]

mae = mean\_absolute\_error(y\_true, y\_pred)

print("MAE:", mae)

# Error 0 (los valores predichos deben ser iguales a los reales)

y\_pred\_zero\_error = [[0.5, 1], [-1, 1], [7, -6]]

mae\_zero = mean\_absolute\_error(y\_true, y\_pred\_zero\_error)

print("MAE con error 0:", mae\_zero)

# Ajuste para obtener un MAE de 0.33

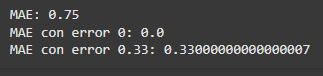
# Sumamos diferencias absolutas de 0.33 en cada posición para la predicción

y\_pred\_033\_error = [[0.83, 1.33], [-0.67, 1.33], [7.33, -5.67]]

mae\_033 = mean\_absolute\_error(y\_true, y\_pred\_033\_error)

print("MAE con error 0.33:", mae\_033)

#### **2.1.2 Ejecución**



### **4.1 Enunciado**

# Ejer 3: Crear dos vectores/arrays de valores ciertos y valores previstos, y calcular el mse.

# ¿Qué valores deberían de tener el array pred para que el error fuese 0? ¿y para que error fuese de 0.25?

# Nota: Es muy interesante comparar los resultados al aplicaer mae y mse sobre el mismo conjunto de datos. Podrás obervar

# que mse "penaliza más" (es decir, el número que se obtiene es mayor) con las diferencias entre los valores originales y los predichos que mae

#### **4.1.1 Código**

from sklearn.metrics import mean\_squared\_error

# Ejemplo: Valores reales y previstos

y\_true = [3, -0.5, 2, 7]

y\_pred = [2.5, 0.0, 2, 8]

# Calcular el MSE

mse = mean\_squared\_error(y\_true, y\_pred)

print("MSE:", mse)

# ¿Qué valores deberían de tener el array pred para que el error fuese 0?

# Error 0 (los valores predichos deben ser iguales a los relaes)

y\_pred\_zero\_error = [3, -0.5, 2, 7]

mse\_zero = mean\_squared\_error(y\_true, y\_pred\_zero\_error)

print("MSE con error 0:", mse\_zero)

# ¿Y para que error fuese de 0.25?

# Ejemplo para error 0.25 (promedio de los cuadrados de las diferencias entre los valores reales y previstos debe ser 0.25)

y\_pred\_025\_error = [3, 0.5, 2, 7] # Una pequeña diferencia

mse\_025 = mean\_squared\_error(y\_true, y\_pred\_025\_error)

print("MSE con error 0.25:", mse\_025)

# Comparación Mae y Mse no pedida

# Comparación

y\_pred\_example = [2.5, 0.0, 2, 8]

mae\_example = mean\_absolute\_error(y\_true, y\_pred\_example)

mse\_example = mean\_squared\_error(y\_true, y\_pred\_example)

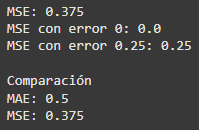
print("")

print("Comparación")

print("MAE:", mae\_example)

print("MSE:", mse\_example)

#### **4.1.2 Ejecución**



### **5.1 Enunciado**

# Ejer 4: Crear dos matrices/arrays de valores ciertos y valores previstos, y calcular el mae.

# ¿Qué valores deberían de tener el array pred para que el error fuese 0? ¿y para que error fuese de 0.33?

#### **5.1.1 Código**

from sklearn.metrics import mean\_absolute\_error

# Crear dos matrices/arrays de valores ciertos y valores previstos

y\_true = [[1, 2], [3, 4], [5, 6]]

y\_pred = [[1.5, 2.5], [2.5, 4.5], [5, 6.5]]

# Calcular el MAE

mae = mean\_absolute\_error(y\_true, y\_pred)

print("MAE:", mae)

# Error 0 (los valores predichos deben ser iguales a los relaes)

y\_pred\_zero\_error = [[1, 2], [3, 4], [5, 6]]

mae\_zero = mean\_absolute\_error(y\_true, y\_pred\_zero\_error)

print("MAE con error 0:", mae\_zero)

# Valores modificados para alcanzar un error promedio de 0.33

# El MAE se calcula promediando las diferencias absolutas entre los elementos de y\_true y y\_pred.

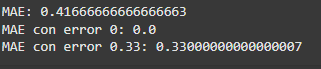
# Para que el MAE sea 0.33, la suma de estas diferencias dividida por el número total de elementos debe ser igual a 0.33.

y\_pred\_033\_error = [[1.33, 2.33], [3.33, 4.33], [5.33, 6.33]]

mae\_033 = mean\_absolute\_error(y\_true, y\_pred\_033\_error)

print("MAE con error 0.33:", mae\_033)

#### **5.1.2 Ejecución**



## **6 Explicación**

# Error de porcentaje absoluto medio - mape

from sklearn.metrics import mean\_absolute\_percentage\_error

y\_true = [3, -0.5, 2, 7]

y\_pred = [2.5, 0.0, 2, 8]

mean\_absolute\_percentage\_error(y\_true, y\_pred)

–

Resultado: 0.32

### **6.1 Enunciado**

# Ejer 5: Crear dos vectores/arrays de valores ciertos y valores previstos, y calcular el mape.

# ¿Qué valores deberían de tener el array pred para que el error fuese 0?

#### **6.1.1 Código**

from sklearn.metrics import mean\_absolute\_percentage\_error

# Vectores: Valores reales y previstos

y\_true = [3, -0.5, 2, 7]

y\_pred = [2.5, 0.0, 2, 8]

# Calcular el MAPE

mape = mean\_absolute\_percentage\_error(y\_true, y\_pred)

print("MAPE:", mape)

# Error 0 (los valores predichos deben ser iguales a los relaes)

# Valores previstos que producen MAPE = 0

y\_pred\_zero\_error = [3, -0.5, 2, 7]

# Calcular MAPE con error 0

mape\_zero = mean\_absolute\_percentage\_error(y\_true, y\_pred\_zero\_error)

print("MAPE con error 0:", mape\_zero)

#### **6.1.2 Ejecución**



### **7.1 Enunciado**

# Ejer 6: Crear dos matrices/arrays de valores ciertos y valores previstos, y calcular el MAPE.

# ¿Qué valores deberían de tener el array pred para que el error fuese 0?

#### **7.1.1 Código**

from sklearn.metrics import mean\_absolute\_percentage\_error

# Matrices: Valores reales y previstos

y\_true = [[1, 2], [3, 4], [5, 6]]

y\_pred = [[1.1, 2.1], [2.9, 4.1], [5.2, 5.9]]

# Calcular el MAPE

mape = mean\_absolute\_percentage\_error(y\_true, y\_pred)

print("MAPE:", mape)

# Error 0 (los valores predichos deben ser iguales a los relaes)

y\_pred\_zero\_error = [[1, 2], [3, 4], [5, 6]]

# Calcular MAPE con error 0

mape\_zero = mean\_absolute\_percentage\_error(y\_true, y\_pred\_zero\_error)

print("MAPE con error 0:", mape\_zero)

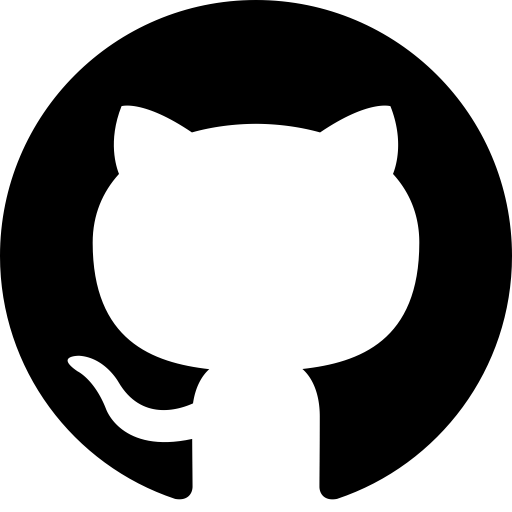
#### **7.1.2 Ejecución**



## **8. Conclusión**

El MAE mide el error absoluto promedio entre los valores reales y los predichos, en las mismas unidades de los datos. El MAPE calcula el error relativo promedio como un porcentaje de los valores reales, lo que es útil para comparaciones en diferentes escalas. Mientras el MAE es robusto a valores pequeños, el MAPE puede ser sensible a valores cercanos a cero.

## **9. Github y Colab**

[](https://github.com/AdrianYArmas/IaBigData/tree/main/SNS/3%20%20-%20Algoritmos%20y%20herramientas%20para%20el%20aprendizaje%20supervisado%20/3.1%20%20M%C3%A9tricas%20de%20error) [](https://colab.research.google.com/drive/1DkV-7FUAh_5PYH9paJO9IMQbdzRQy21F?usp=sharing)