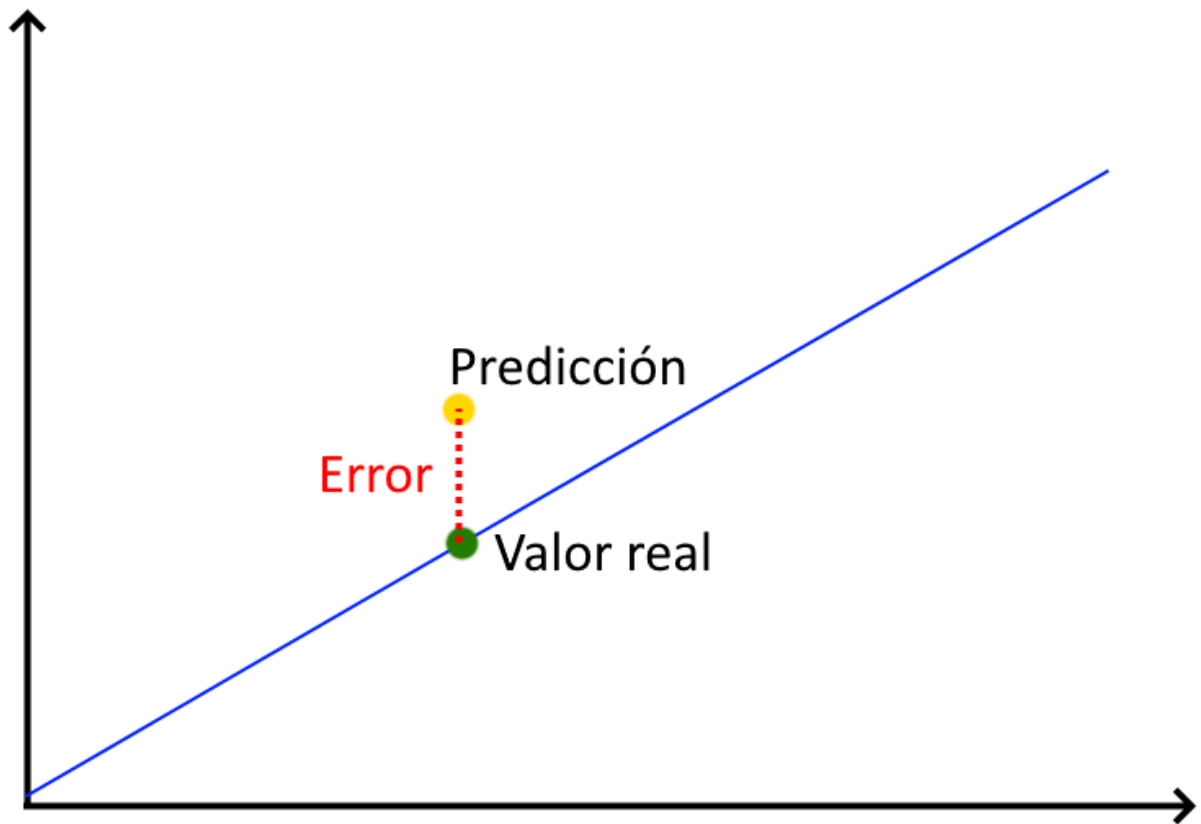


Métricas de error



Adrián Yared Armas de la Nuez



Contenido

1. Enunciado.....	2
2 Explicación.....	2
2.1 Enunciado.....	2
2.1.1 Código.....	3
2.1.2 Ejecución.....	3
3 Explicación.....	3
3.1 Enunciado.....	4
3.1.1 Código.....	4
2.1.2 Ejecución.....	4
4.1 Enunciado.....	5
4.1.1 Código.....	5
4.1.2 Ejecución.....	6
5.1 Enunciado.....	6
5.1.1 Código.....	6
5.1.2 Ejecución.....	7
6 Explicación.....	7
6.1 Enunciado.....	7
6.1.1 Código.....	7
6.1.2 Ejecución.....	8
7.1 Enunciado.....	8
7.1.1 Código.....	8
7.1.2 Ejecución.....	9
8. Conclusión.....	9
9. Github y Colab.....	9

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>

Mape:

$$M = \frac{1}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right|$$

M = error absoluto medio porcentual

n = número de veces que se produce la iteración de la suma

At = valor real

Ft = valor previsto

Mae:

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

MAE= error absoluto medio

yi = predicción

xi = valor verdadero

n= número total de datos

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
# 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

```
MAE: 0.5
MAE con error 0: 0.0
MAE con error 0.25: 0.25
```

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

```
MAE: 0.75
MAE con error 0: 0.0
MAE con error 0.33: 0.33000000000000007
```

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 aplicar mae y mse sobre el mismo conjunto de datos. Podrás observar

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 reales)
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

```
MSE: 0.375
MSE con error 0: 0.0
MSE con error 0.25: 0.25

Comparación
MAE: 0.5
MSE: 0.375
```

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 reales)
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

```
MAE: 0.4166666666666667
MAE con error 0: 0.0
MAE con error 0.33: 0.33000000000000007
```

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 reales)
# 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

```
MAPE: 0.3273809523809524
MAPE con error 0: 0.0
```

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 reales)
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

```
MAPE: 0.044166666666666674
MAPE con error 0: 0.0
```

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

