

Medición del error

```
In [1]: 1 import pandas as pd
2 import numpy as np
3 import math
4 import matplotlib.pyplot as plt
5 import time
```

```
In [2]: 1 df = pd.read_csv('Datos.csv')
2 df = df[['reales', 'predichos']][0:100]
3 print(df.size)
4 df.head(3)
```

200

```
Out[2]:
```

	reales	predichos
0	113.0	66.0
1	141.0	90.0
2	118.0	111.0

Mean Square Error (MSE)

Es el método más comúnmente utilizado para poder predecir el error.

También se utiliza su raíz cuadrada, llamadas root means-squared error

$$MSE = \frac{(p_1 - a_1)^2 + \dots + (p_n - a_n)^2}{n}$$

$$RMSE = \sqrt{\frac{(p_1 - a_1)^2 + \dots + (p_n - a_n)^2}{n}}$$

```
In [3]: 1 numero_datos = len(df)
```

```
In [4]: 1 inicio = time.time()
2 sumatoria = 0
3
4 #función para calcular la diferencia entre p y a, y elevar el resultado al cuadrado
5 def diferencia_cuadrada_potencia(p_i, a_i):
6     return (p_i - a_i)**2
7
8 #Calcular para todas las instancias
9 for i in range(0, numero_datos):
10     sumatoria += diferencia_cuadrada_potencia(df['predichos'][i], df['reales'][i])
11
12 MSE = sumatoria/numero_datos
13 RMSE = math.sqrt(MSE)
14 fin = time.time()
15
16 print('MSE = ', MSE)
17 print('RMSE = ', RMSE)
18
19 T_MSE_RMSE = (fin - inicio)
```

MSE = 707.6

RMSE = 26.600751869073175

Mean Absolute Error (MAE)

Es una medida que solo promedia las magnitudes de cada error de manera individual sin tener en cuenta su signo.

Los outliers no son exagerados.

$$MAE = \frac{|p_1 - a_1| + \dots + |p_n - a_n|}{n}$$

```
In [5]: 1 sumatoria = 0
2 inicio = time.time()
3 #función para calcular el valor absoluto de la diferencia entre p y a
4 def diferencia_abs(p_i, a_i):
5     return abs(p_i - a_i)
6
7 #Calcular para todas las instancias
```

```

8 for i in range(0,numero_datos):
9     sumatoria += diferencia_abs(df['predichos'][i],df['reales'][i])
10
11 MAE = sumatoria/numero_datos
12 fin = time.time()
13
14 print('MAE = ',MAE)
15 T_MAE = (fin-inicio)

```

MAE = 19.16

Relative Squared Error (RSE)

Se promedian todos los valores obtenidos con respecto a los valores de entrenamiento (p), por lo que RSE toma todos los errores al cuadrado y los normaliza dividiendo todos los errores del predictor

$$RSE = \frac{(p_i - a_i)^2 + \dots + (p_n - a_n)^2}{(a_i - \bar{a})^2 + \dots + (a_n - \bar{a})^2}$$

where $\bar{a} = \frac{1}{n} \sum a_i$

```

In [6]: 1 inicio = time.time()
2         # a_negada = promedio de a
3         promedio_a_n = df['reales'].sum()
4         promedio_a_n /= numero_datos
5
6
7         dividendo = 0
8         divisor = 0
9
10
11         #Calcular para todas las instancias
12         for i in range(0,numero_datos):
13             dividendo += diferencia_cuadrada_potencia(df['predichos'][i],df['reales'][i])
14
15
16         for j in range(0,numero_datos):
17             divisor += diferencia_cuadrada_potencia(df['reales'][j],promedio_a_n)
18
19
20
21 RSE = dividendo/divisor
22 RRSE = math.sqrt(RSE)
23 fin = time.time()
24
25 print('RSE = ',round(RSE,4))
26 print('RRSE = ',round(RRSE,4))
27 T_RSE = (fin-inicio)

```

RSE = 0.4185

RRSE = 0.6469

Relative Absolute Error (RAE)

Es el valor error absoluto, normalizado de la misma manera que con MAE

$$RAE = \frac{|p_i - a_i| + \dots + |p_n - a_n|}{|a_i - \bar{a}| + \dots + |a_n - \bar{a}|}$$

```

In [7]: 1 inicio = time.time()
2
3         dividendo = 0
4         divisor = 0
5
6         #Calcular para todas las instancias
7         for i in range(0,numero_datos):
8             dividendo += diferencia_abs(df['predichos'][i],df['reales'][i])
9
10
11         for j in range(0,numero_datos):
12             divisor += diferencia_abs(df['reales'][j],promedio_a_n)
13
14 RAE = dividendo/divisor
15 fin = time.time()
16
17 print('RAE = ',RAE)
18 T_RAE = (fin-inicio)

```

RAE = 0.6333381814335386

Correlation Coefficient (CC)

Mide la correlación estadística entre los valores actuales (a) y los valores predichos (p).

El coeficiente de correlación varía de 1 para un resultado de correlación perfecto a un 0, donde no existe correlación o -1 cuando la correlación es perfecta pero negativa.

Correlation coefficient:

$$\frac{S_{pa}}{\sqrt{S_p S_a}}$$

where

$$S_{pa} = \frac{\sum(p-\bar{p})(a-\bar{a})}{n-1}$$

```
In [8]: 1 inicio = time.time()
2 # p_negada = promedio de p
3 promedio_p_n = df['predichos'].sum()
4 promedio_p_n /= numero_datos
```

```
In [9]: 1 def suma_producto_spa(p, p_n, a, a_n):
2     return (p-p_n)*(a-a_n)
3
4 sumatoria_spa = 0
5
6 for i in range(0,numero_datos):
7     sumatoria_spa += suma_producto_spa(df['predichos'][i],promedio_p_n,df['reales'][i],promedio_a_n)
8
9 s_pa = sumatoria_spa/(numero_datos-1)
```

$$S_p = \frac{\sum(p-\bar{p})^2}{n-1}$$

```
In [10]: 1 p_prom = 0
2 for j in range(0,numero_datos):
3     p_prom += diferencia_cuadrada_potencia(df['predichos'][j],promedio_p_n)
4 s_p = p_prom/(numero_datos-1)
```

$$S_a = \frac{\sum(a-\bar{a})^2}{n-1}$$

```
In [11]: 1 a_prom = 0
2 for k in range(0,numero_datos):
3     a_prom += diferencia_cuadrada_potencia(df['reales'][k],promedio_a_n)
4 s_a = a_prom/(numero_datos-1)
```

$$\text{Correlation Coefficient (CC)} = \frac{S_{pa}}{\sqrt{(S_p S_a)}}$$

```
In [12]: 1 CC = s_pa/( math.sqrt(s_p*s_a) )
2 fin = time.time()
3
4 print('CC = ',CC)
5 T_CC = (fin-inicio)
```

CC = 0.8158317676525696

Medida de medición del error

• En general, se utiliza de manera indistinta la medición del error o por defecto se utiliza **RMSE**. En realidad, no existe una receta para siempre utilizar un método en particular, sino que se dependerá de los datos, la aplicación, etcétera.

- Se tiene que preguntar: ¿Qué se está tratando de minimizar?
- ¿Cuál es el costo que representa el resultado (y los cálculos) de cada uno de los métodos
- ¿Cuándo es más útil utilizar un método con respecto a los otros?

```
In [13]: 1 print('Valor de los errores calculados')
2 print('Mean Square Error: ',MSE)
3 print('Roore Mean Square Error: ',RMSE)
4 print('Mean Absolute Error: ',MAE)
5 print('Relative Squared Error: ',RSE)
6 print('Root Relative Squared Error: ',RRSE)
7 print('Relative Absolute Error: ',RAE)
8 print('Correlation Coefficient: ',CC)
```

Valor de los errores calculados
Mean Square Error: 707.6
Roore Mean Square Error: 26.600751869073175
Mean Absolute Error: 19.16
Relative Squared Error: 0.4184889556116051
Root Relative Squared Error: 0.6469072233416513
Relative Absolute Error: 0.6333381814335386
Correlation Coefficient: 0.8158317676525696

No se esta tratando de minimizar nada, lo que se trata es de calcular el error en los algoritmos, mediante los diferentes métodos, para posteriormente tratar de minimizar la señal de error algo así como un sistema de control.

Costo que representa:

```
In [14]: 1 print('Tiempos')
2 print('MSE y RMSE: ', T_MSE_RMSE)
3 print('MAE: ', T_MAE)
4 print('RSE: ', T_RSE)
5 print('RAE: ', T_RAE)
6 print('CC: ', T_CC)
```

```
Tiempos
MSE y RMSE: 0.0019998550415039062
MAE: 0.0019979476928710938
RSE: 0.002000093460083008
RAE: 0.0019996166229248047
CC: 0.06251144409179688
```

El dataset es pequeño y se muestran los tiempos de ejecución los cuales son muy bajos, con un conjunto de datos más grande se podría apreciar mejor la duración de estos, pero se espera que MSE , RMSE , MAE tengan menor tiempo de ejecución ya que se definen menos operaciones.

Su utilidad correspondera a lo que se busca calcular.

MSE: proporciona una función cuadrática de pérdida en la medida en que eleva al cuadrado y, posteriormente, promedia los varios errores; lo cual le da mucho más peso a los grandes errores (valores atípicos) que a los más pequeños.

MAE: Este criterio es similar al coeficiente RMSE. Sin embargo, es más robusto ya que es menos sensible a valores extremos que el MSE.

Todas las medidas de distancia (MSE, RMSE y MAE) son equivalentes y permiten cuantificar cómo las soluciones aproximadas coinciden con los datos simulados. Un valor pequeño para estos criterios significa que el modelo estimado puede predecir los valores de la respuesta del modelo más complejo.

RSE: El error cuadrático relativo (RSE) es relativo a lo que habría sido si se hubiera utilizado un predictor simple. Más específicamente, este simple predictor es solo el promedio de los valores reales. Por lo tanto, el error cuadrático relativo toma el error cuadrático total y lo normaliza dividiéndolo por el error cuadrático total del predictor simple. Se puede comparar entre modelos cuyos errores se miden en las diferentes unidades.

RAE: Es una forma de medir el rendimiento de un modelo predictivo. No debe confundirse con el error relativo, que es una medida general de precisión o exactitud para instrumentos como relojes, reglas o escalas. Se expresa como una razón, comparando un error medio (residual) con los errores producidos por un modelo trivial o ingenuo. Un buen modelo de pronóstico producirá un índice cercano a cero; Un modelo deficiente (uno que es peor que el modelo ingenuo) producirá una proporción mayor que uno.

CC: Para dos variables, la fórmula compara la distancia de cada dato puntual respecto a la media de la variable y utiliza esta comparación para decirnos hasta qué punto la relación entre las variables se ajusta a una línea imaginaria trazada entre los datos. A esto nos referimos cuando decimos que la correlación examina las relaciones lineales.

```
1 https://www.sciencedirect.com/topics/engineering/mean-absolute-error
```

```
In [15]: 1 plt.plot(df['reales'], '-bo', label = 'Reales')
2 plt.plot(df['predichos'], '-ro', label = 'Predicciones')
3 plt.legend()
4 plt.grid()
```

