Informática y comunicaciones

ies azarquiel

Curso especialización en inteligencia artificial y big data

Análisis de datos

Profesor: Sebastián Rubio Valero

Septiembre 2025

Big Data

1 Escalonamiento de datos

El escalamiento de datos (o data scaling) es el proceso de ajustar el rango o la magnitud de las variables en un conjunto de datos, de modo que todas contribuyan de forma equilibrada a los análisis o modelos que se construyan con ellas, es decir poner todas la variables (características) en la misma escala para que ninguna domine o distorsione los resultados. Supongamos que tenemos los siguientes datos de la tabla, donde puedes ver que las tres características tienen diferente magnitudes

Table 1: Placeholder Caption

Variable	**Rango**
Ingreso anual	10.000 – 200.000
Edad	18 – 80
Tasa de interés	0,01 – 0,2

Si se alimenta un modelo con estos datos tal cual, el algoritmo puede darle más peso a las variable con valores grandes (como el ingreso) aunque no sean los más importantes.
El escalonamiento evita esto, permitiendo que los algoritmos aprendan o analicen las relaciones de formas más equilibradas.
Para cada valor x de una característica X:

$$x' = \frac{x - X_{\min}}{X_{\max} - X_{\min}} \tag{1}$$

Donde x', es el valor escalado entre 0 y 1
También se puede ajustar a un rango salida usando la versión generalizada

$$x' = (x - X_{min}).\frac{b - a}{X_{\max} - X_{\min}} + a \tag{2}$$

donde a y b son los límites deseados.

1.1 Cuando usar el escalonamiento simple

Se suele utilizar en aquellos modelos que son sensibles a la escala de datos, como:

- Modelos basados en gradiente descendiente
- k-Means
- KNN

- Redes Neuronales.

Es un técnica de transformación que es sensible a los outliers, puede comprimir el resto de los datos hacia un rango muy estrecho

2 Estandarización

También se le llama Z-score normalization, es una técnica de escalonamiento que transforma los datos que tengan una $Media(\mu) = 0$ y una desviación estándar$(\sigma) = 1$, no tiene un rango como en el anterior, además es menos sensible a los outliers La fórmula es:

$$x' = \frac{x - \mu}{\sigma} \tag{3}$$

2.1 Cuando usar

Se usan en muchos modelos que suponen que los datos están centrados y con varianza constante.

- Regresión lineal y logística
- Máquina Vector Soporte (SVM)
- Modelos basados en gradientes (GBM,XGBoot)
- Redes Neuronales

3 Normalización

A veces se le confunde con el escalonamiento Min-Max. Consiste en ajustar los valores numéricos de los datos para que se encuentren en una misma escala o rango común, con el fín de que todas las variables contribuyan de forma equitativa al entrenamiento del modelo, es decir, pone todas la variables al mismo nivel, eliminando diferencias de magnitud o unidades.
Se usa principalmente cuando trabajamos con vectores de características, entonces el objetivo es ajustar la longitud del vector, no su rango numérico Esta es la normalización Euclídea

$$x' = \frac{x}{\sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}} \tag{4}$$

Hace que la longitud del vector sea 1, preservando direcciones pero igualando magnitudes.
La normalización Manhattan

$$x' = \frac{x}{|x_1| + |x_2| + \cdots + |x_n|} \tag{5}$$

Es muy usada en el procesamiento de lenguaje natural y en el análisis de texto e imágenes

Ejemplo

Sea el vector $\vec{x} = [3, 4]$
La norma euclídea

$$\|\vec{x}\|_2 = \sqrt{3^2 + 4^2} = \sqrt{9 + 16} = 5 \tag{6}$$

Dividimos cada componente por la norma:

$$\vec{x'} = [3/5, 4/5] = [0.6, 0.8] \tag{7}$$

$$\sqrt{0.6^2 + 0.8^2} = 1 \tag{8}$$

3.1 Cuando usar la normalización

Es ideal para modelos basados en distancia como KNN, K-means, redes neuronales. Hay que tener en cuenta que es muy sensible a los outliers

4 Binarización

Se usa especilamente cuando trabajamos con algoritmos que necesitan entradas numéricas discretas (0 y 1) o cuando queremos simplificar la información de una variable contínua o categórica. Convierte los valores númericos o categóricos en valores bianrios, es decir convierte datos como "si/no", "verdadero/falso","presente/ausente" en cero o uno
Define un valor umbral (threshold), y según él, cada valor se convierte en 0 o 1

$$x' = \begin{cases} 1, & \text{si } x > \text{umbral} \\ 0, & \text{si } x \leq \text{umbral} \end{cases} \tag{9}$$

4.1 Cuando usar

- Quiere transformar una variable contínua en dos clases (p.e "bajo riesgo/alto riesgo"

- Trabajas con modelos lineales o lógicos que esperan datos binarios (p.e regresión lineal, Naive Bayes).
- Quiere simplificar una variable para detección de patrones o clasificación binaria

5 Box-Cox

Esta técnica de transformación se aplica a variables numéricas positivas, y está orientada a reducir:

- Reducir el sesgo de los datos
- Estabilizar la varianza
- Aproximar la distribución de los datos a una forma normal

Para cada $y > 0$,la transformación depende de un parámetro λ

$$y(\lambda) = \begin{cases} \frac{y^\lambda - 1}{\lambda}, & \text{si } \lambda \neq 0 \\ ln(y), & \text{si } \lambda = 0 \end{cases} \tag{10}$$

Donde:

- y es el valor original (debe ser positivo)
- λ es un parámetro de transformación (determina el tipo de cambio)

El parámetro λ controla cuánto se deforma la variable, se elige de forma que los datos transformado sean lo más cercanos a una distribución normal

Table 2: Relación entre el valor de λ y la Transformación de Box-Cox

Valor de λ	**Transformación aproximada**	**Efecto**
2	Cuadrado	Aumenta valores grandes
1	Sin cambio	Igual a los datos originales
0.5	Raíz cuadrada	Reduce asimetría derecha
0	Logarítmica	Reduce asimetría derecha (caso especial)
-1	Inversa $(1/y)$	Aumenta simetría cuando hay sesgo fuerte

Ejemplo

Si tus datos tienen una cola larga hacia la derecha (p.e, ingresos o tiempos de espera), Box-Cox con $\lambda \approx 0 o 0.5$ puede comprimir esos valores altos, haciendo la distribución más simétrica

5.1 Cuando usar

Es conveniente usar esta técnica de transformación cuando:

- Tus variables solo tienen valores positivos
- Los datos presentan asimetría
- Necesitas que los datos cumplan supuestos de normalidad (p.e, regresión lineal)
- Buscas mejorar el rendimiento de modelos que son sensibles a escalas y distribución (p.e, regresión, SVM,LDA)

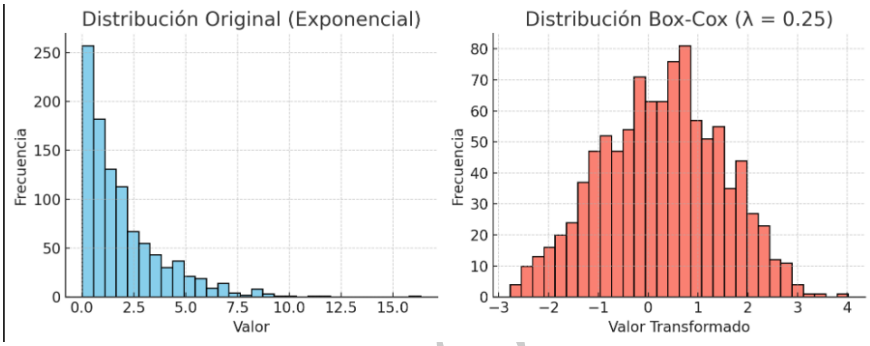

Figure 1: Enter Caption

6 Yeo-Johnson

Es igual de Box-Cox, pero además acepta valores negativos.
Para cada valor de y, la función se define:

$$y(\lambda) = \begin{cases} \frac{y^{\lambda}-1}{\lambda}, & \text{si } y \geq 0, \lambda \neq 0 \\ ln(y+1), & \text{si } y \geq 0, \lambda = 0 \\ -\frac{(-y+1)^{2-\lambda}-1}{2-\lambda}, & \text{si } y < 0, \lambda \neq 2 \\ -ln(-y+1), & \text{si } y < 0, \lambda = 2 \end{cases} \tag{11}$$

Es decir, para valores positivos, se comporta como Box-Cox, pero para valores negativos, aplica una transformación similar pero invertida.
El parámetro λ ajusta cuánto se corrige la asimetría. Se elige automáticamente (normalmente por máxima verosimilitud) para que los datos transformados sean lo más gaussiando posible.

Table 3: Relación entre el valor de λ y la Transformación de Box-Cox (Variante)

Valor de λ	**Transformación aproximada**	**Efecto**
2	Raíz cuadrada inversa	Comprime valores grandes
1	Sin cambio	Igual a los datos originales
0	Logarítmica (para positivos)	Reduce asimetría derecha
-1	Inversa	Simetriza distribuciones muy sesgadas

6.1 Cuando usar

- Tus datos tienen además de los valores positivos, tienen ceros y valores negativos
- Necesitas mejorar la normalidad o estabilizar la varianza
- Trabajas con modelos sensibles a la distribución (regresión lineal, SVM, etc)
- Quiere un preprocesamiento automático y reversible

No es conveniente usarlo en árboles, o descenso del gradiente