AULA ESTADÍSTICA

Serie Monográfica: Normalización -Estandarización de Variables

Autor: Jorge Lorenzo

Noviembre 2024

Normalización-Estandarización de Variables

¿Qué es la normalización?

La normalización o escalado de variables es una técnica de transformación de datos que se realiza para situar todas las características de las mismas en un rango similar, en función de la técnica estadística que se haya elegido para su procesamiento. Cuando se trabaja con modelos complejos, el proceso de normalización de variables mejora su rendimiento, ya que ayuda a que los algoritmos de aprendizaje automático, especialmente aquellos que dependen de la distancia (como k-NN y SVM), funcionen de manera más efectiva. Al tener todas las variables en una escala similar, se evita que las variables con rangos más amplios dominen el cálculo de la distancia. La normalización también facilita la convergencia en los algoritmos de optimización en modelos que utilizan las distancias como parte principal de la técnica de análisis (tal el caso de la regresión logística o las redes neuronales). Esto se debe a que el espacio de búsqueda se vuelve más homogéneo, permitiendo que el algoritmo encuentre soluciones óptimas más rápidamente. Al aplicar métodos de normalización, se reduce la influencia de valores atípicos. Esto es particularmente útil en conjuntos de datos donde los valores extremos pueden distorsionar los resultados, mejorando así la interpretabilidad al lograr que el rango de valores de las variables sea más fácilmente comparable. Por ejemplo, al escalar las variables a un rango común (como [0, 1]), se hace más sencilla la comparación de los efectos de diferentes variables estudiada. En el caso de la preparación para técnicas de análisis multivariado, tales como el análisis de componentes principales (PCA) o distintas técnicas de regresión, se asume que las variables están en la misma escala. En tal sentido, la normalización permite que los datos cumplan con esta suposición, mejorando la calidad y la validez de los resultados.

Estos beneficios hacen que la normalización de variables sea un paso fundamental en el preprocesamiento de datos en análisis estadísticos y en el aprendizaje automático. Para comprender diferentes procedimientos de normalización se ofrece un ejemplo

sencillo sobre el cual se aplicaron distintas técnicas de normalización[1]. En cada caso se ofrece un script para reproducir el ejemplo con el uso del software RStudio[2].

Crear un data.frame en RStudio

Un data.frame en RStudio es una estructura que permite almacenar datos tabulares de manera similar a una hoja de cálculo o una tabla en una base de datos. Es una de las estructuras más utilizadas en R para el análisis de datos y tiene las siguientes características:

Estructura de filas y columnas: Un data.frame se compone de filas y columnas, donde cada columna puede contener un tipo de dato diferente (números, cadenas de texto, etc.), todas las entradas en una misma columna son del mismo tipo y definen una variable.

Acceso a los datos: Se pueden acceder a los elementos de un data.frame utilizando notación de corchetes, nombres de columnas o funciones específicas. Una vez cargado el data.frame en el software se puede comenzar con los análisis pertinentes.

Dimensiones: Para verificar las dimensiones de un data.frame (número de filas y columnas), se utiliza la función `dim()`. También puedes utilizar `nrow()` y `ncol()` para obtener el número de filas y columnas, respectivamente.

Funciones de manipulación: R ofrece una amplia variedad de funciones para manipular y analizar data.frames, incluyendo `subset()` para seleccionar subconjuntos de datos, `merge()` para combinar varios data.frames, y funciones del paquete `dplyr` (tales como `filter()`, `select()`, `mutate()`, etc.), para realizar operaciones más complejas en la limpieza y optimización de la base de datos con la que se quiere trabajar. En todos los casos, el software posee un glosario de ayuda que explica cada una de estas funciones, por lo que enumerarlas a todas en redundante.

Facilidad de uso: Los data.frames son intuitivos y fáciles de usar, lo que los convierte en una herramienta fundamental para la mayoría de los análisis de datos en R. Permiten organizar y gestionar grandes conjuntos de datos de manera eficiente. Asimismo, son exportables para ser utilizados en otros softwares distintos.

El ejemplo que se va a remixar requiere crear un data.frame que contiene siete variables y cinco casos. La matriz es sencilla por defecto dado que la finalidad de este ejemplo es

[1] La base de datos utilizada en este ejemplo originalmente se usa para ejemplificar los procedimientos con Phyton; en este caso se adaptó el ejemplo original para trabajar con RStudio.
[2] RStudio es un entorno de desarrollo integrado (IDE) para el lenguaje de programación R, dedicado a la computación estadística. Para más detalles del software se recomienda su sitio web: https://posit.co/downloads/

hacer accesible diferentes técnicas de normalización. En la siguiente tabla se muestra el código para aplicar en RStudio y generar el data.frame de trabajo:

Código 1: Generar el conjunto de datos en RStudio

```
> # Cargar las bibliotecas necesarias
> library(dplyr)
> library(scales)
>
> # Crear la matriz con los datos
> arr <- matrix(c('M', 81.4, 82.2, 44, 6.1, 120000, 'no',
+                 'M', 75.2, 86.2, 40, 5.9, 80000, 'no',
+                 'F', 80.0, 83.2, 34, 5.4, 210000, 'yes',
+                 'F', 85.4, 72.2, 46, 5.6, 50000, 'yes',
+                 'M', 68.4, 87.2, 28, 5.11, 70000, 'no'),
+               nrow = 5, byrow = TRUE)
>
> # Convertir la matriz a un data frame
> df <- as.data.frame(arr, stringsAsFactors = FALSE)
>
> # Asignar nombres a las columnas
> colnames(df) <- c('genero', 'hsc_p', 'ssc_p', 'edad', 'altura', 'salario', 'enferme
dad')
>
> # Convertir las columnas numéricas a tipo numérico
> df$hsc_p <- as.numeric(df$hsc_p)
> df$ssc_p <- as.numeric(df$ssc_p)
> df$age <- as.numeric(df$age)
> df$height <- as.numeric(df$height)
> df$salary <- as.numeric(df$salary)
>
> # Mostrar el data frame
> print(df)
```

El resultado se muestra en la siguiente tabla; las variables siguen la siguiente nomenclatura: Género: (M= masculino; F= femenino); hsc_p: puntaje obtenido en el examen de finalización de secundaria (*Higher Secondary Certificate*), que equivale al nivel de educación secundaria superior (como el bachillerato en muchos países). Generalmente, se trata del rendimiento académico en los últimos años de la escuela secundaria; ssc_p: es el porcentaje o puntaje obtenido en el examen de secundaria (Secondary *School Certificate*), que es un nivel de educación secundaria (generalmente, hasta los 16 años). Representa el rendimiento académico en la educación secundaria básica. Estos valores se utilizan a menudo en análisis de datos educativos o laborales para analizar el rendimiento académico de las personas en diferentes etapas de su educación. Las restantes variables de la tabla son, la edad, altura, salario actual, y una pregunta que indaga sobre el estado de salud (si sufre alguna enfermedad).

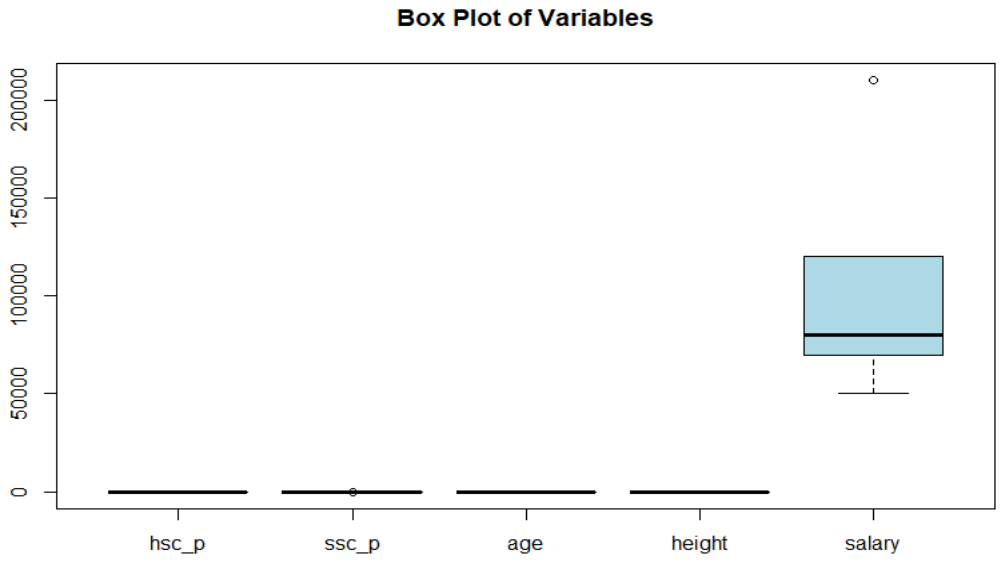
Como se aprecia, la tabla combina distintas variables, cada una medida en su propia escala y con valores muy diferentes entre sí. Por lo tanto, se trata de un buen caso para aplicar la técnica de normalización.

Tabla 1: datos del ejemplo normalización

Genero	hsc_p	ssc_p	age	height	salary	disease
M	81.4	82.2	44	6.10	120000	no
M	75.2	86.2	40	5.90	80000	no
F	80.0	83.2	34	5.40	210000	yes
F	85.4	72.2	46	5.60	50000	yes
M	68.4	87.2	28	5.11	70000	no

A continuación, y con los datos actuales, esto es, sin haber operado sobre ellos ninguna transformación, se muestra un gráfico de cajas (box-plot) para cada una de las variables. Se aprecia que la variable salario sobresale en su representación. Esto se debe a que, sus valores dominan sobre el eje de la Y, y por tanto las otras variables aparecen distorsionadas en la gráfica. En otras palabras, tomando una base común en el eje de las Y, el salario extiende su rango hasta cubrir el valor extremo de del salario, subrepresentando el resto de las variables. Claramente no es posible comparaciones sobre los valores originales de las variables y por tanto se necesita de una transformación adecuada.

Gráfico 1: representación en diagrama de cajas de las variables sin transformación



Transformación Escalar Min-Max

Esta técnica de transformación, se basa en los valores mínimo y máximo que puede tomar una variable. A cada valor de x se le resta el mínimo valor que asuma x en la variable y luego se divide por la diferencia entre el valor máximo y el mínimo. Así, el rango de valores originales, queda escalado al rango [0;1].

$$x_{escalado} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Según se aprecia en la ecuación siguiente, cuando x es igual al valor mínimo el numerador es cero, y por lo tanto el cociente es cero; si x es igual al valor máximo, tanto el numerador como el denominador son iguales y en tal caso el resultado será uno. Cualquier valor de x que no sea el mínimo o el máximo queda dentro del intervalo 0; 1. Se aprecia que esta transformación puede aplicarse a variables continuas y numéricas, y no puede utilizarse con valores binarios o categóricos.

Código 2: escalado Min-Max para la variable hcs_p

```
# Cargar la biblioteca necesaria
> library(scales)
>
> # Asegurarse de que la columna 'hsc_p' sea numérica
> df$hsc_p <- as.numeric(df$hsc_p)
>
> # Aplicar el escalado Min-Max (entre 0 y 1)
> df$hsc_p <- rescale(df$hsc_p, to = c(0, 1))
>
> # Mostrar el data.frame transformado
> print(df)
```

A continuación, se muestra la tabla 2 con la variable hcs_p normalizada por el método Min-Max. Nótese que los valores originales ahora están comprendidos en el rango [0;1].

Tabla 2: variable hsc_p normalizada con la técnica Min-Max

gender	hsc_p	ssc_p	age	height	salary	_disease
M	0.7647059	82.2	44	6.10	120,000	no
M	0.4000000	86.2	40	5.90	80,000	no
F	0.6823529	83.2	34	5.40	210,000	yes
F	1.0000000	72.2	46	5.60	50,000	yes
M	0.0000000	87.2	28	5.11	70,000	no

El rango de transformación de la técnica Min-Max que usualmente se utiliza es el que se halla en el intervalo 0 y 1; no obstante, pueden usarse otros rangos distintos según la naturaleza de reescalamiento del problema en análisis.

Estandarización

La estandarización es quizás la técnica de escalamiento más conocida y utilizada en estadística. Se basa en el cálculo de las puntuaciones Z. En este caso, los valores de la variable original quedan circunscriptos al rango [-1;1]. A diferencia de la transformación Min-Max, la puntuación estándar Z utiliza el promedio y la desviación estándar, tal como se muestra en la siguiente ecuación:

$$Z = \frac{x - \bar{x}}{de}$$

Así, cada puntuación Z es el resultado de la sustracción de cada valor de x respecto al promedio, dividido la desviación estándar. Toda transformación Z de los datos originales, da por resultado una nueva distribución cuyo promedio es cero y su desviación estándar es uno.

Código 3: transformación Z para la variable ssc_p

```
# Asegurarse de que la columna 'ssc_p' sea numérica
df$ssc_p <- as.numeric(df$ssc_p)

# Aplicar el escalado estándar (z-score: (x - mean) / sd)
df$ssc_p <- scale(df$ssc_p)

# Mostrar el data.frame transformado
> print(df)
```

A continuación, en la tabla 3 se muestra la variable ssc_p expresada en valores Z. Nótese que ahora las dos variables referidas a rendimiento académico, tienen valores comprendidos en rangos similares.

Tabla 3: transformación Z de la variable ssc_p

gender	hsc_p	ssc_p	age	height	salary	disease
M	0.7647059	0.0000000	44	6.10	120,000	no
M	0.4000000	0.6713451	40	5.90	80,000	no
F	0.6823529	0.1678363	34	5.40	210,000	yes
F	1.0000000	-1.6783627	46	5.60	50,000	yes
M	0.0000000	0.8391814	28	5.11	70,000	no

Normalización

En el contexto de normalización, los términos L1 y L2 se refieren a diferentes tipos de normas matemáticas utilizadas para medir la magnitud de vectores. Estas normas son fundamentales en la normalización de datos, especialmente en el aprendizaje automático, porque permiten ajustar las variables manteniendo relaciones relativas, pero restringiendo su magnitud.

Norma L1: es la suma de los valores absolutos de los elementos de un vector y se expresa mediante la siguiente ecuación:

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|$$

Al aplicar la ecuación se divide cada elemento del vector entre la suma de los valores absolutos, de modo que la suma de los valores normalizados sea 1. La normalización L1 a menudo se utiliza cuando queremos destacar la dispersión relativa de las variables, ya que preserva la estructura de los datos esparcidos.

Norma L2: es la raíz cuadrada de la suma de los cuadrados de los elementos del vector y se expresa mediante la siguiente ecuación:

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$$

En esta segunda ecuación se divide cada elemento del vector entre su norma L2, de modo que la magnitud total del vector (según la distancia euclidiana) sea igual a 1. Esta normalización es adecuada cuando buscamos estabilidad matemática en modelos que trabajan con vectores, como SVM o redes neuronales.

Existen diferencias clave entre normalización y escalado Min-Max. Aunque tanto la normalización (mediante normas L1 o L2) como el escalado Min-Max resultan en valores que pueden estar en el rango [0, 1] (dependiendo de los datos), tienen objetivos y métodos distintos. La Normalización (L1 oL2), se centra en ajustar las relaciones dentro de un vector o fila para garantizar que su magnitud siga ciertas restricciones (como suma 1 o longitud 1). No depende de los valores mínimos o máximos absolutos de la columna.

Para el ejemplo que estamos desarrollando utilizaremos la L2, que, como se ha mostrado da como resultado valores entre 0 y 1; la ecuación simplificada de esta operación se muestra a continuación:

$$x_{norm} = \frac{x}{\|x\|_2}$$

La expresión matemática representa un proceso de normalización L2, donde x es el vector de datos que se desea normalizar. Puede representar un conjunto de características, como las puntuaciones de los estudiantes, medidas físicas, etc., que en este caso será la altura. La expresión $\|x\|_2$ es la norma L2 del vector x, también conocida como la longitud o magnitud del vector. Su cálculo se mostró más arriba. De este modo, para la variable altura tendremos un escalado uniforme; todos los elementos se escalan para que la magnitud total del vector sea 1. La normalización L2 mantiene la dirección del vector original. Esto significa que, aunque se modifique la magnitud, el vector sigue apuntando en la misma dirección.

Código 4: normalización L2 para la variable altura (height)

```
# Asegurarse de que la columna 'height' sea numérica
df$height <- as.numeric(df$height)

# Normalizar la columna 'height' utilizando la norma L2
# Primero, calcular la norma L2
l2_norm <- sqrt(sum(df$height^2))

# Normalizar la columna dividiendo cada elemento por la norma L2
df$height <- df$height / l2_norm

# Mostrar el data.frame transformado
print(df)
```

En la tabla 4 se muestra la variable altura (height) normalizada. La altura ahora asume valores comparables a las otras variables en que se han aplicado transformaciones.

Tabla 4: normalización L2 para la variable altura (height)

gender	hsc_p	ssc_p	age	height	salary	disease
M	0.7647059	0.0000000	44	0.4842916	120,000	no
M	0.4000000	0.6713451	40	0.4684132	80,000	no
F	0.6823529	0.1678363	34	0.4287171	210,000	yes
F	1.0000000	-1.6783627	46	0.4445956	50,000	yes
M	0.0000000	0.8391814	28	0.4056934	70,000	no

Escalamiento Robusto

El escalamiento robusto se utiliza para describir una técnica de normalización que es resistente a valores atípicos (outliers), ya que se basa en estadísticas robustas como la

mediana y el rango intercuartílico (IQR), en lugar de la media y la desviación estándar. El término robusto hace referencia a esta cualidad, pues durante el reescalado de los datos, éstos no se ven significativamente afectados por desviaciones como valores atípicos o distribuciones no normales. En este método se escalan los datos en el rango entre el 1er cuartil y el 3er cuartil, es decir, entre el rango del 25º cuantil y el 75º cuantil. Este rango también se denomina rango intercuartílico. La ecuación para la transformación de los datos mediante el escalamiento robusto se muestra a continuación:

$$E = \frac{x_i - Q_1(x)}{Q_1(x) - Q_3(x)}$$

En lenguajes de programación como Python existe una función para realizar esta tarea conocido como `RobustScaler()`, en a RStudio, se puede usar la función `scale()` de R, ajustando los parámetros para que se asemeje a la funcionalidad de `RobustScaler()`, que se basa en la mediana y el rango intercuartílico. Sin embargo, en R no hay un equivalente directo a `RobustScaler()`, por lo que es necesario calcular manualmente la mediana y el rango intercuartílico.

Código 5: normalización robusta para la variable salario

```
# Asegurarse de que la columna 'salary' sea numérica
df$salary <- as.numeric(df$salary)

# Calcular la mediana y el rango intercuartílico (IQR)
median_salary <- median(df$salary, na.rm = TRUE)
IQR_salary <- IQR(df$salary, na.rm = TRUE)

# Aplicar el escalado robusto
df$salary <- (df$salary - median_salary) / IQR_salary

# Mostrar el data.frame transformado
print(df)
```

Con este código se opera la conversión numérica de la variable salario, luego se calcula la mediana y el rango intercuartílico para la columna salario, que son necesarios para el escalado robusto. Finalmente, se aplica el escalado robusto restando la mediana y dividiendo por el rango intercuartilar (IQR). En la siguiente tabla, se muestra el resultado de la transformación.

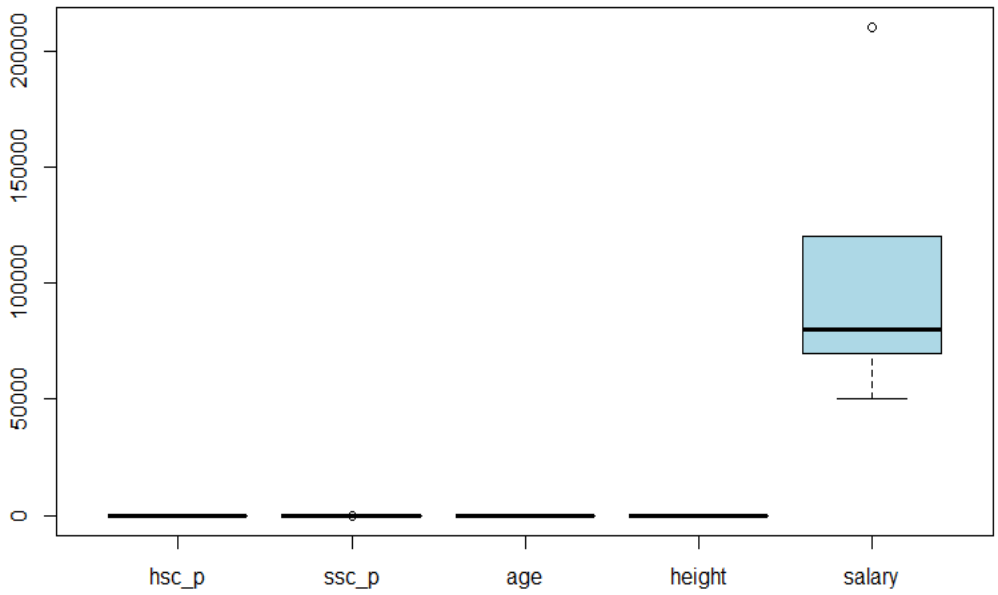
Tabla 5: normalización robusta para la variable salario

gender	hsc_p	ssc_p	age	height	salary	disease
M	0.7647059	0.0000000	44	0.4842916	0.8	no
M	0.4000000	0.6713451	40	0.4684132	0.0	no
F	0.6823529	0.1678363	34	0.4287171	2.6	yes
F	1.0000000	-1.6783627	46	0.4445956	-0.6	yes
M	0.0000000	0.8391814	28	0.4056934	-0.2	no

Resumen del proceso

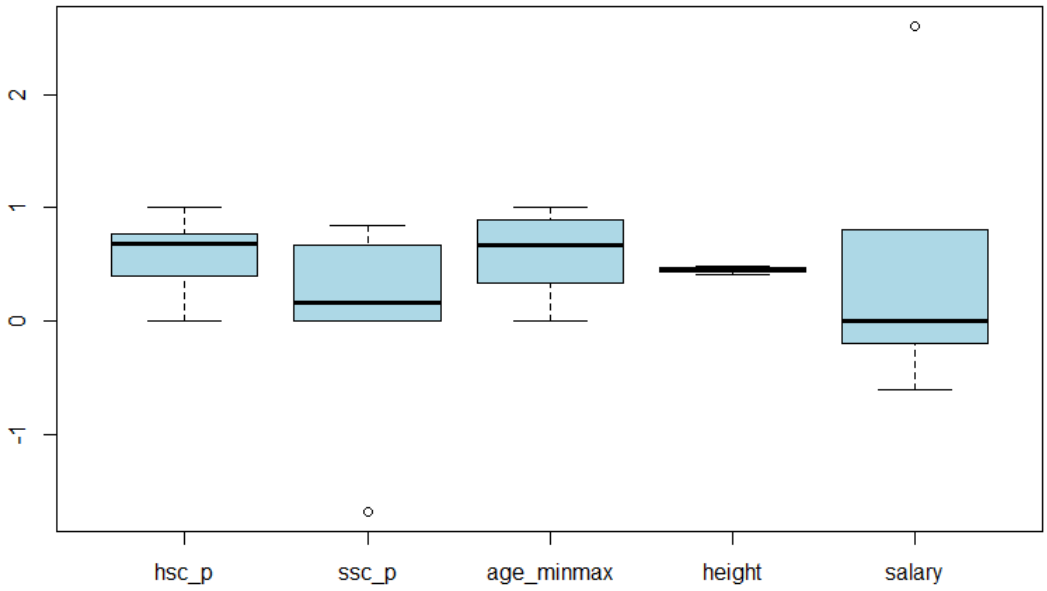
En el data.frame con que comenzamos este ejemplo, existen siete variables, dos de la cuales son nominales (género y si padece una enfermedad), las restantes cinco son de tipo métricas. Tal como se mostró al principio del artículo, un gráfico de cajas de todas ellas sin haber operado ninguna transformación, provocaba que la variable salario apareciera sobrerrepresentada, dado que consumía la mayor parte de los valores del eje de las *Y*. Cualquier comparación en esta situación resulta cuanto menos engorrosa, cuando no, inútil.

Gráfico 2: representación en diagrama de cajas de las variables sin transformación



A lo largo de los ejemplos, hemos visto cómo pueden aplicarse distintas técnicas de reescalamiento para lograr que las variables queden comprendidas entre valores similares, sea en el dominio [0;1], o en [-1;1]. De este modo al graficar nuevamente mediante un diagrama de cajas, vemos que la posición de las mismas y la referencia al eje Y, hace menos problemática su comparación (en este caso la variable edad se transformó con el método Min-Max).

Gráfico 3: representación en diagrama de cajas de las variables transformadas



Normalización y Estandarización

Tanto la normalización como la estandarización son técnicas utilizadas para transformar los datos a una escala común, pero tienen fines distintos y se utilizan para distintos propósitos. La normalización es una técnica que escala los datos a un rango común, normalmente entre 0 y 1, para evitar que las variables con un rango de valores muy amplio dominen el modelo de análisis. Se aplica cuando las variables fueron medidas en unidades muy diferentes (en este ejemplo, edad y salario), y ayuda a eliminar el efecto de las diferentes unidades de medida. También se aplica cuando un modelo de análisis de datos es sensible a las escalas de medición, v.g. las redes neuronales, en tal caso la normalización ayuda a estabilizar el proceso de entrenamiento y el rendimiento del modelo. La técnica de normalización Min-Max es un buen ejemplo de cómo restringir el rango de datos de una variable métrica continua.

La estandarización es una técnica que transforma los datos para que tengan una media de 0 y una desviación estándar de 1, lo que facilita la comparación y combinación de

características. Resulta útil cuando las variables tienen distribuciones diferentes, pues ayuda transformar diferentes distribuciones en una distribución común, lo que facilita su comparación y combinación. Algunos modelos, como la regresión lineal, asumen la normalidad de las variables y la estandarización facilita el cumplimiento de este supuesto.

La principal diferencia entre normalización y estandarización es que la normalización escala los datos a un rango común, mientras que la estandarización transforma los datos para que tengan una media de 0 y una desviación estándar de 1. La transformación Z que se repasó anteriormente es un buen ejemplo de estandarización.

La normalización se emplea más cuando las variables tienen unidades o escalas diferentes, cuando el modelo de análisis es sensible a las escalas de las variables, y cuando el objetivo es evitar que las variables con rangos grandes dominen el modelo. La estandarización es preferible cuándo las variables tienen distribuciones diferentes, cuando el modelo asume la normalidad de las variables, y cuando el objetivo es evaluar la importancia de cada variable. No obstante, debe tenerse en cuenta que la estandarización (o transformación Z) de una variable no corrige el problema de la no normalidad de los datos. La estandarización es un método de transformación que convierte los valores de una variable a una escala con media 0 y desviación estándar 1, pero no altera la forma de la distribución subyacente de los datos. En este sentido, el principal propósito de la estandarización es ajustar la escala de los datos para hacerlos comparables en términos de magnitudes relativas. Esto es útil, por ejemplo, en algoritmos de aprendizaje automático o análisis estadístico que son sensibles a escalas (como PCA Principal Component Analysis, SVM, Support Vector Machine, etc.).

Por lo dicho, si los datos son sesgados o presentan asimetrías, la distribución estandarizada también será sesgada. Esto significa que un conjunto de datos que no sigue una distribución normal seguirá sin ser normal tras la transformación Z. Entonces, si la normalidad es un requisito para el modelo analítico, es necesario realizar una transformación diferente que busque aproximar los datos a una distribución normal. Por ejemplo: a) Transformaciones logarítmicas (log); b) Transformaciones de raíz cuadrada; c) Transformaciones Box-Cox o Yeo-Johnson, que son más generales y buscan ajustar los datos a la normalidad de manera flexible.

En modelos algorítmicos de aprendizaje automático (ML) la normalidad no es estrictamente necesaria. Sin embargo, la estandarización sigue siendo útil para garantizar que las variables estén en la misma escala. Por lo tanto, la estandarización no corrige la falta de normalidad; simplemente ajusta la escala de los datos. Si se

necesita que los datos sean normales para un análisis, se debe considerar una transformación que altere la forma de la distribución o recurrir a métodos no paramétricos.