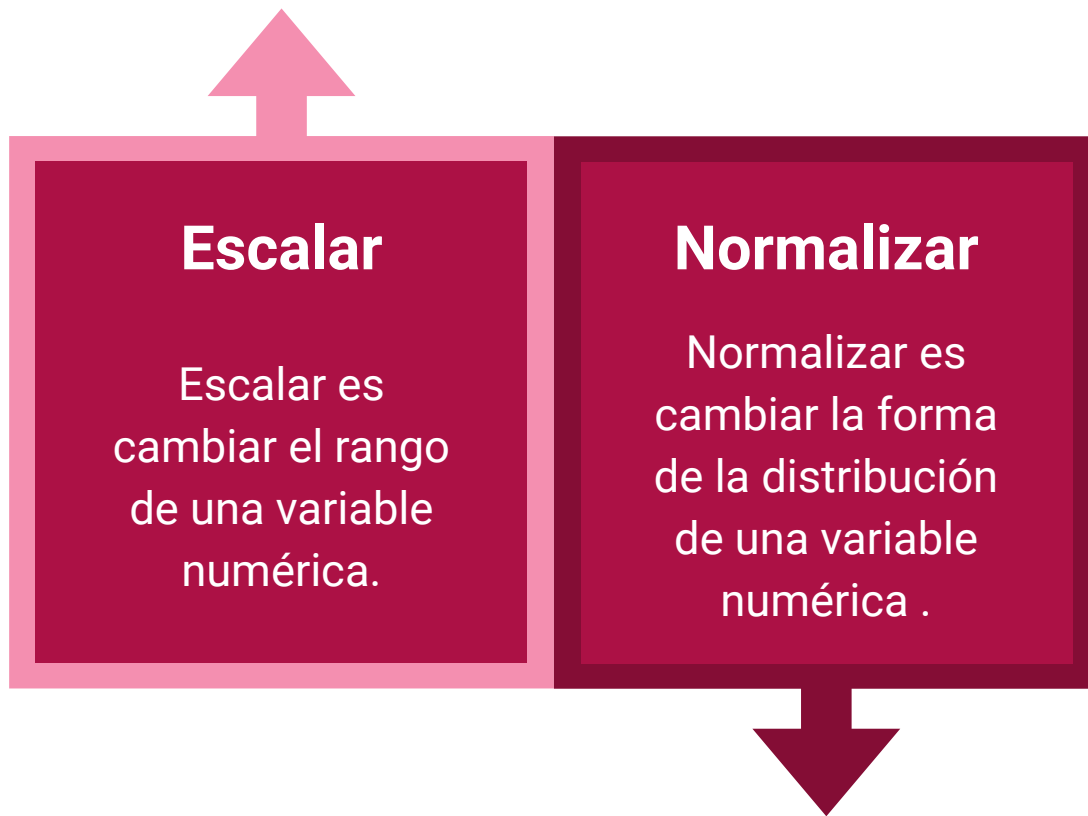


Transformaciones



escalar y normalizar

Escalar vs Normalizar, cuál es la diferencia?



Escalado

- ❖ El escalado implica transformar los datos para que ajusten a una escala determinada, como 0-100 o $[0,1]$.
- ❖ Es importante escalar los datos antes de usar métodos basados en distancias, como Support Vector Machines o K nearest neighbors.
- ❖ Por ejemplo, si se tienen precios de productos en yens y en dolares, un dolar son más de 100 yens, pero estos algoritmos consideran el cambio en una unidad igual para ambas características.
- ❖ Escalando, las variables pasan a ser pesadas de la misma forma
- ❖ Sklearn tiene los métodos MinMaxScaler y MaxAbsScaler

Escalado

- ❖ El escalado implica transformar los datos para que ajusten a una escala determinada, como 0-100 o $[0,1]$.
- ❖ Es importante escalar los datos antes de usar métodos basados en distancias, como Support Vector Machines o K nearest neighbors.
- ❖ Por ejemplo, si se tienen precios de productos en yens y en dolares, un dolar son más de 100 yens, pero estos algoritmos consideran el cambio en una unidad igual para ambas características.
- ❖ Escalando, las variables pasan a ser pesadas de la misma forma
- ❖ Sklearn tiene los métodos MinMaxScaler y MaxAbsScaler

Escalado

- ❖ El escalado implica transformar los datos para que ajusten a una escala determinada, como 0-100 o $[0,1]$.
- ❖ Es importante escalar los datos antes de usar métodos basados en distancias, como Support Vector Machines o K nearest neighbors.
- ❖ Por ejemplo, si se tienen precios de productos en yens y en dolares, un dolar son más de 100 yens, pero estos algoritmos consideran el cambio en una unidad igual para ambas características.
- ❖ Escalando, las variables pasan a ser pesadas de la misma forma
- ❖ Sklearn tiene los métodos MinMaxScaler y MaxAbsScaler

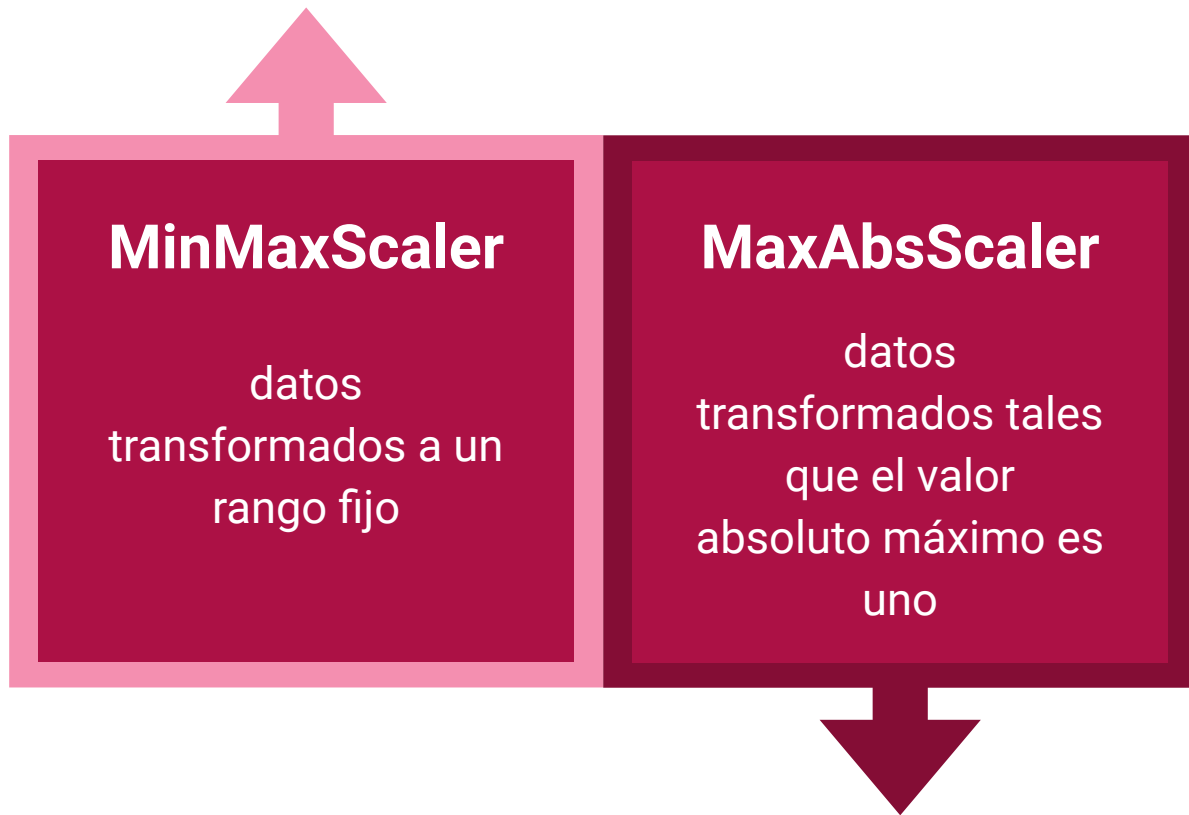
Escalado

- ❖ El escalado implica transformar los datos para que ajusten a una escala determinada, como 0-100 o $[0,1]$.
- ❖ Es importante escalar los datos antes de usar métodos basados en distancias, como Support Vector Machines o K nearest neighbors.
- ❖ Por ejemplo, si se tienen precios de productos en yens y en dolares, un dolar son más de 100 yens, pero estos algoritmos consideran el cambio en una unidad igual para ambas características.
- ❖ Escalando, las variables pasan a ser pesadas de la misma forma
- ❖ Sklearn tiene los métodos `MinMaxScaler` y `MaxAbsScaler`

Escalado

- ❖ El escalado implica transformar los datos para que ajusten a una escala determinada, como 0-100 o $[0,1]$.
- ❖ Es importante escalar los datos antes de usar métodos basados en distancias, como Support Vector Machines o K nearest neighbors.
- ❖ Por ejemplo, si se tienen precios de productos en yens y en dolares, un dolar son más de 100 yens, pero estos algoritmos consideran el cambio en una unidad igual para ambas características.
- ❖ Escalando, las variables pasan a ser pesadas de la misma forma
- ❖ Sklearn tiene los métodos `MinMaxScaler` y `MaxAbsScaler`

MinMaxScaler vs MaxAbsScaler



MinMaxScaler vs MaxAbsScaler

MinMaxScaler La motivación para usar esta escala incluye la solidez contra las desviaciones estándar muy pequeñas en las variables y la preservación de entradas cero en datos raros.

MaxAbsScaler funciona de manera muy similar, pero escala forzando los datos al rango $[-1, 1]$, dividiendo por el valor máximo (absoluto) en cada característica. Está destinado a datos que ya están centrados en cero o datos raros.

Centrar datos raros destruiría la estructura de dispersión de los datos y, por lo tanto, rara vez es algo sensato. Sin embargo, puede tener sentido escalar entradas raras, especialmente si las variables están en escalas diferentes.

MaxAbsScaler fue diseñado específicamente para escalar datos raros y es la forma recomendada de hacerlo

MinMaxScaler vs MaxAbsScaler

MinMaxScaler La motivación para usar esta escala incluye la solidez contra las desviaciones estándar muy pequeñas en las variables y la preservación de entradas cero en datos raros.

MaxAbsScaler funciona de manera muy similar, pero escala forzando los datos al rango $[-1, 1]$, dividiendo por el valor máximo (absoluto) en cada característica. Está destinado a datos que ya están centrados en cero o **datos raros**.

Centrar datos raros destruiría la estructura de dispersión de los datos y, por lo tanto, rara vez es algo sensato. Sin embargo, puede tener sentido escalar entradas raras, especialmente si las variables están en escalas diferentes.

MaxAbsScaler fue diseñado específicamente para escalar datos raros y es la forma recomendada de hacerlo

Standardization

- ❖ La estandarización de conjuntos de datos es un requisito común para muchos estimadores de aprendizaje automático implementados en scikit-learn; podrían comportarse mal si las características individuales no se parecen más o menos a datos distribuidos normal patrón: **Gaussiano con media cero y varianza unitaria.**
- ❖ En la práctica, a menudo ignoramos la forma de la distribución y simplemente transformamos los datos para centrarlos eliminando el valor medio de cada característica, luego la escalamos dividiendo las características no constantes por su desviación estándar usando **StandardScaler**
- ❖ Si sus datos contienen muchos valores atípicos, es probable que el escalado utilizando la media y la varianza de los datos no funcione muy bien. En estos casos, puede usar **RobustScaler** como reemplazo directo. Utiliza estimaciones más sólidas para el centro y rango de sus datos.

Standardization

- ❖ La estandarización de conjuntos de datos es un requisito común para muchos estimadores de aprendizaje automático implementados en scikit-learn; podrían comportarse mal si las características individuales no se parecen más o menos a datos distribuidos normal patrón: Gaussiano con media cero y varianza unitaria.
- ❖ En la práctica, a menudo ignoramos la forma de la distribución y simplemente transformamos los datos para centrarlos eliminando el valor medio de cada característica, luego la escalamos dividiendo las características no constantes por su desviación estándar usando **StandardScaler**
- ❖ Si sus datos contienen muchos valores atípicos, es probable que el escalado utilizando la media y la varianza de los datos no funcione muy bien. En estos casos, puede usar **RobustScaler** como reemplazo directo. Utiliza estimaciones más sólidas para el centro y rango de sus datos.

Standardization

- ❖ La estandarización de conjuntos de datos es un requisito común para muchos estimadores de aprendizaje automático implementados en scikit-learn; podrían comportarse mal si las características individuales no se parecen más o menos a datos distribuidos normal patrón: Gaussiano con media cero y varianza unitaria.
- ❖ En la práctica, a menudo ignoramos la forma de la distribución y simplemente transformamos los datos para centrarlos eliminando el valor medio de cada característica, luego la escalamos dividiendo las características no constantes por su desviación estándar usando **StandardScaler**
- ❖ Si sus datos contienen muchos valores atípicos, es probable que el escalado utilizando la media y la varianza de los datos no funcione muy bien. En estos casos, puede usar **RobustScaler** como reemplazo directo. Utiliza estimaciones más sólidas para el centro y rango de sus datos.

Normalization

- ❖ Hay dos tipos de transformaciones disponibles: **transformaciones de cuantiles y transformaciones de potencia**. Tanto las transformaciones de cuantiles como las de potencia se basan en transformaciones monótonas de las características y, por lo tanto, preservan el rango de los valores a lo largo de cada característica.
- ❖ Las transformadas de cuantiles colocan todas las características en la misma distribución deseada según la fórmula $G^{-1}(F(X))$ donde F es la función de distribución acumulativa de la característica y G^{-1} la función de cuantiles de la distribución de salida deseada G

Normalization

- ❖ Esta fórmula utiliza los dos hechos siguientes: (i) si X es una variable aleatoria con una función de distribución acumulativa continua F , entonces $F(X)$ se distribuye uniformemente en $[0,1]$ (ii) si U es una variable aleatoria con distribución en $[0,1]$ entonces $G^{-1}(U)$ tiene distribución G
- ❖ .Al realizar una transformación de rango, una transformación de cuantiles suaviza distribuciones inusuales y está menos influenciada por valores atípicos que los métodos de escala. Sin embargo, distorsiona las correlaciones y distancias dentro y entre entidades.
- ❖ `QuantileTransformer` proporciona una transformación no paramétrica para asignar los datos a una distribución uniforme con valores entre 0 y 1:

Normalization

- ❖ Esta fórmula utiliza los dos hechos siguientes: (i) si X es una variable aleatoria con una función de distribución acumulativa continua F , entonces $F(X)$ se distribuye uniformemente en $[0,1]$ (ii) si U es una variable aleatoria con distribución en $[0,1]$ entonces $G^{-1}(U)$ tiene distribución G
- ❖ .Al realizar una transformación de rango, una transformación de cuantiles suaviza distribuciones inusuales y está menos influenciada por valores atípicos que los métodos de escala. Sin embargo, distorsiona las correlaciones y distancias dentro y entre entidades.
- ❖ `QuantileTransformer` proporciona una transformación no paramétrica para asignar los datos a una distribución uniforme con valores entre 0 y 1:

Normalization

- ❖ Esta fórmula utiliza los dos hechos siguientes: (i) si X es una variable aleatoria con una función de distribución acumulativa continua F , entonces $F(X)$ se distribuye uniformemente en $[0,1]$ (ii) si U es una variable aleatoria con distribución en $[0,1]$ entonces $G^{-1}(U)$ tiene distribución G
- ❖ .Al realizar una transformación de rango, una transformación de cuantiles suaviza distribuciones inusuales y está menos influenciada por valores atípicos que los métodos de escala. Sin embargo, distorsiona las correlaciones y distancias dentro y entre entidades.
- ❖ `QuantileTransformer` proporciona una transformación no paramétrica para asignar los datos a una distribución uniforme con valores entre 0 y 1:

Power transformations

- ❖ Las transformaciones de potencia son una familia de transformaciones paramétricas que tienen como objetivo mapear datos de cualquier distribución lo más cerca posible de una distribución gaussiana.
- ❖ La transformación de Yeo-Johnson y la de Box-Cox son

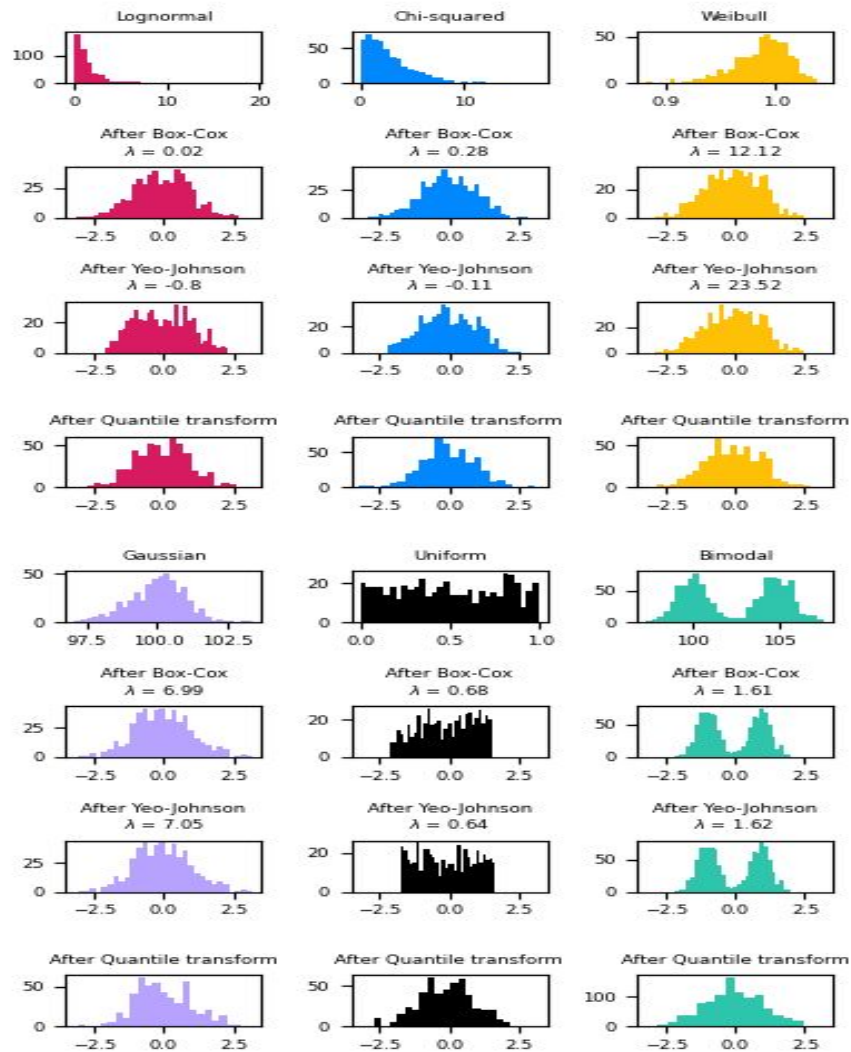
$$x_i^{(\lambda)} = \begin{cases} [(x_i + 1)^\lambda - 1]/\lambda & \text{if } \lambda \neq 0, x_i \geq 0, \\ \ln(x_i + 1) & \text{if } \lambda = 0, x_i \geq 0 \\ -[(-x_i + 1)^{2-\lambda} - 1]/(2 - \lambda) & \text{if } \lambda \neq 2, x_i < 0, \\ -\ln(-x_i + 1) & \text{if } \lambda = 2, x_i < 0 \end{cases} \quad x_i^{(\lambda)} = \begin{cases} \frac{x_i^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \ln(x_i) & \text{if } \lambda = 0. \end{cases}$$

Power transformations

```
bc = PowerTransformer(method='box-cox')
```

```
yj = PowerTransformer(method='yeo-johnson')
```

```
qr = QuantileTransformer(output_distribution='normal',  
    random_state=0)
```



Normalization (llevar a norma unitaria)

- ❖ La normalización es el proceso de escalar muestras individuales para tener una norma unitaria.
- ❖ Este proceso puede ser útil si planea utilizar una forma cuadrática como el producto punto o cualquier otro núcleo para cuantificar la similitud de cualquier par de muestras.
- ❖ La función `normalize` proporciona una manera rápida y fácil de realizar esta operación en un único conjunto de datos similar a una matriz, ya sea usando las normas l_1 , l_2 o \max

Normalization (llevar a norma unitaria)

- ❖ La normalización es el proceso de escalar muestras individuales para tener una norma unitaria.
- ❖ Este proceso puede ser útil si planea utilizar una forma cuadrática como el producto punto o cualquier otro núcleo para cuantificar la similitud de cualquier par de muestras.
- ❖ La función `normalize` proporciona una manera rápida y fácil de realizar esta operación en un único conjunto de datos similar a una matriz, ya sea usando las normas l_1 , l_2 o \max

Normalization (llevar a norma unitaria)

- ❖ La normalización es el proceso de escalar muestras individuales para tener una norma unitaria.
- ❖ Este proceso puede ser útil si planea utilizar una forma cuadrática como el producto punto o cualquier otro núcleo para cuantificar la similitud de cualquier par de muestras.
- ❖ La función `normalize` proporciona una manera rápida y fácil de realizar esta operación en un único conjunto de datos similar a una matriz, ya sea usando las normas l_1 , l_2 o max