Leave a **Lasting** Impact

# Aprendizado de máquina - Escala

&lt; Anterior

Próximo &gt;

## Recursos de escala

Quando seus dados têm valores diferentes e até unidades de medida diferentes, pode ser difícil compará-los. O que é quilogramas comparado a metros? Ou altitude em relação ao tempo?

A resposta para este problema é a escala. Podemos dimensionar os dados em novos valores que são mais fáceis de comparar.

Dê uma olhada na tabela abaixo, é o mesmo conjunto de dados que usamos no [capítulo de regressão múltipla](#), mas desta vez a coluna de **volume** contém valores em *litros* em vez de  $cm^3$  (1,0 em vez de 1000).

O arquivo é apenas para fins de teste, você pode baixá-lo aqui: [cars2.csv](#)

Carro	Modelo	Volume	Peso	CO2
Toyota	Aygo	1,0	790	99
Mitsubishi	Estrela do Espaço	1.2	1160	95
Skoda	Citigo	1,0	929	95
Fiat	500	0,9	865	90
Mini	Tanoeiro	1,5	1140	105
VW	Acima!	1,0	929	105

Skoda	Fábia	1,4	1109	90
Mercedes	Uma aula	1,5	1365	92
Ford	Festa	1,5	1112	98
Audi	A1	1,6	1150	99

It can be difficult to compare the volume 1.0 with the weight 790, but if we scale them both into comparable values, we can easily see how much one value is compared to the other.

There are different methods for scaling data, in this tutorial we will use a method called standardization.

The standardization method uses this formula:

$$z = (x - u) / s$$

Where **z** is the new value, **x** is the original value, **u** is the mean and **s** is the standard deviation.

If you take the **weight** column from the data set above, the first value is 790, and the scaled value will be:

$$(790 - 1292.23) / 238.74 = -2.1$$

If you take the **volume** column from the data set above, the first value is 1.0, and the scaled value will be:

$$(1.0 - 1.61) / 0.38 = -1.59$$

Now you can compare -2.1 with -1.59 instead of comparing 790 with 1.0.

You do not have to do this manually, the Python sklearn module has a method called `StandardScaler()` which returns a Scaler object with methods for transforming data sets.

## Example

Scale all values in the Weight and Volume columns:

```
import pandas
from sklearn import linear_model
from sklearn.preprocessing import StandardScaler
scale = StandardScaler()
```

```
df = pandas.read_csv("cars2.csv")

X = df[['Weight', 'Volume']]

scaledX = scale.fit_transform(X)

print(scaledX)
```

## Result:


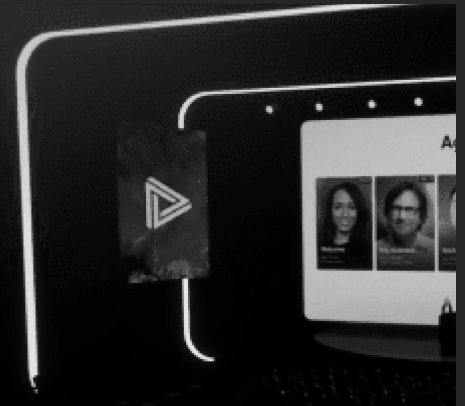
Note that the first two values are -2.1 and -1.59, which corresponds to our calculations:

```
[[-2.10389253 -1.59336644]
 [-0.55407235 -1.07190106]
 [-1.52166278 -1.59336644]
 [-1.78973979 -1.85409913]
 [-0.63784641 -0.28970299]
 [-1.52166278 -1.59336644]
 [-0.76769621 -0.55043568]
 [ 0.3046118  -0.28970299]
 [-0.7551301  -0.28970299]
 [-0.59595938 -0.0289703 ]
 [-1.30803892 -1.33263375]
 [-1.26615189 -0.81116837]
 [-0.7551301  -1.59336644]
 [-0.16871166 -0.0289703 ]
 [ 0.14125238 -0.0289703 ]
 [ 0.15800719 -0.0289703 ]
 [ 0.3046118  -0.0289703 ]
 [-0.05142797  1.53542584]
 [-0.72580918 -0.0289703 ]
 [ 0.14962979  1.01396046]
 [ 1.2219378  -0.0289703 ]
 [ 0.5685001   1.01396046]
 [ 0.3046118   1.27469315]
 [ 0.51404696 -0.0289703 ]
 [ 0.51404696  1.01396046]
 [ 0.72348212 -0.28970299]
 [ 0.8281997   1.01396046]
 [ 1.81254495  1.01396046]
 [ 0.96642691 -0.0289703 ]
 [ 1.72877089  1.01396046]
 [ 1.30990057  1.27469315]
 [ 1.90050772  1.01396046]
```

```
[ -0.23991961 -0.0289703 ]  
[  0.40932938 -0.0289703 ]  
[  0.47215993 -0.0289703 ]  
[  0.4302729   2.31762392]]
```

[Run example »](#)

ADVERTISEMENT

 TouchcastLeave a **Lasting** Impact

## Predict CO2 Values

The task in the [Multiple Regression chapter](#) was to predict the CO2 emission from a car when you only knew its weight and volume.

When the data set is scaled, you will have to use the scale when you predict values:

### Example

Preveja a emissão de CO2 de um carro de 1,3 litro que pesa 2300 kg:

```
import pandas  
from sklearn import linear_model  
from sklearn.preprocessing import StandardScaler  
scale = StandardScaler()  
  
df = pandas.read_csv("cars2.csv")  
  
X = df[['Weight', 'Volume']]
```

```
y = df['CO2']

scaledX = scale.fit_transform(X)

regr = linear_model.LinearRegression()
regr.fit(scaledX, y)

scaled = scale.transform([[2300, 1.3]])

predictedCO2 = regr.predict([scaled[0]])
print(predictedCO2)
```

## Resultado:

```
[107.2087328]
```

Executar exemplo »

◀ Anterior

Próximo ▶

PROPAGANDA



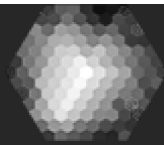
NOVO

Acabamos de lançar  
vídeos do W3Schools



Explorar agora

SELETOR DE CORES



Obtenha a certificação  
completando  
um curso hoje!



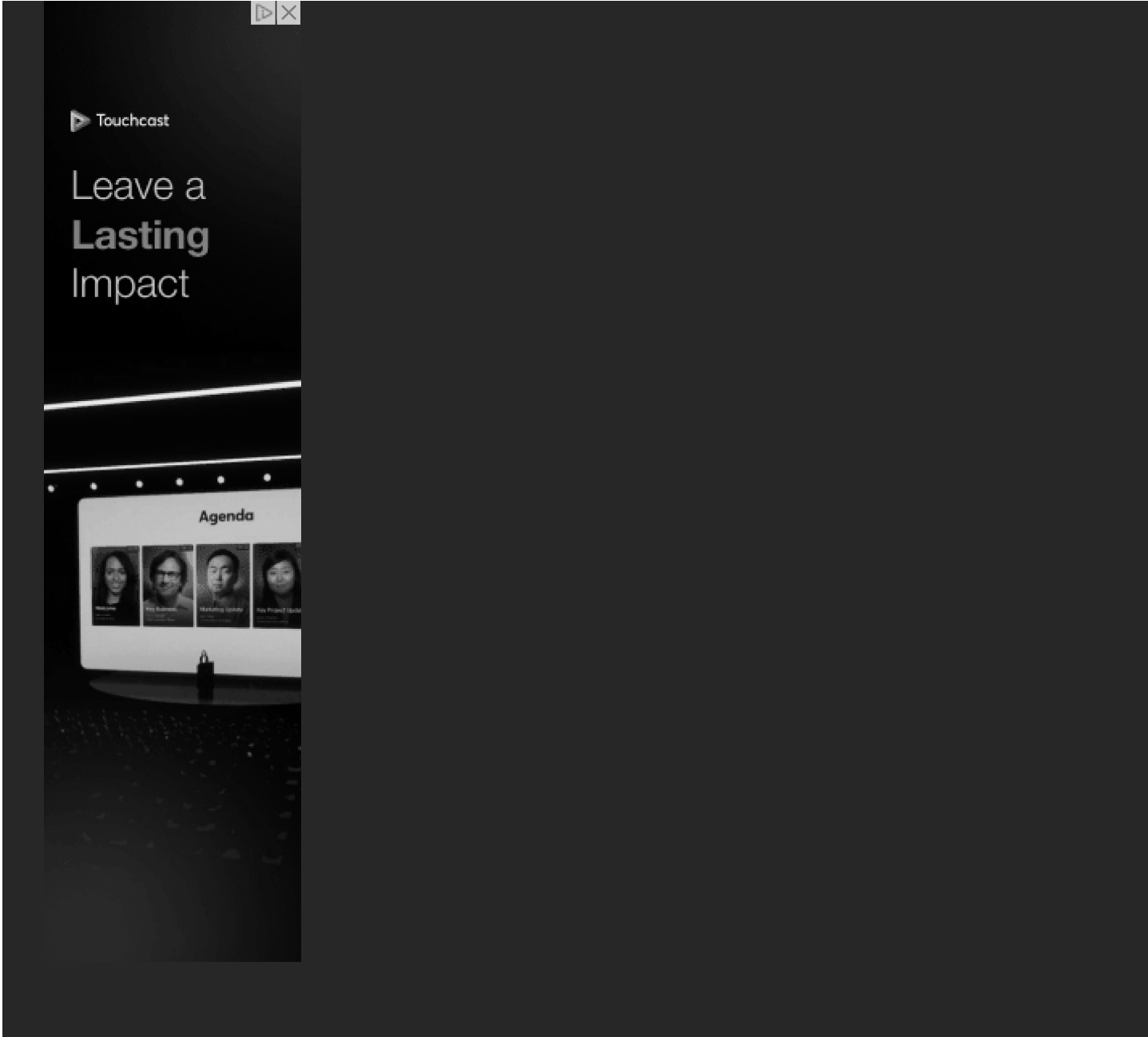
iniciar

JOGO DE CÓDIGO



Jogar um jogo

PROPAGANDA

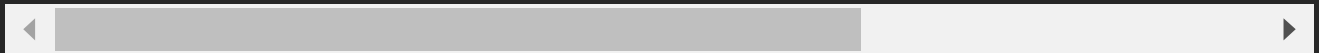
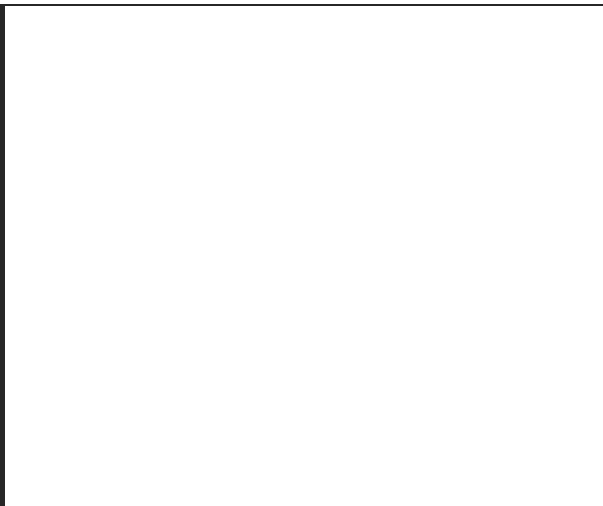


PROPAGANDA



PROPAGANDA



[Reportar erro](#)[Fórum](#)[Sobre](#)[Comprar](#)

## Principais tutoriais

[Tutorial HTML](#) [Tutorial CSS](#) [Tutorial JavaScript](#)  
[How To Tutorial](#) [SQL Tutorial](#) [Python Tutorial](#)  
[W3.CSS Tutorial](#) [Bootstrap Tutorial](#) [PHP Tutorial](#)  
[Java Tutorial](#) [C++ Tutorial](#) [jQuery](#)

## Principais referências

[HTML Reference](#) [CSS Reference](#) [JavaScript Reference](#)  
[SQL Reference](#) [Python Reference](#) [W3.CSS Reference](#)  
[Bootstrap Reference](#) [PHP Reference](#) [HTML Colors](#)  
[Java Reference](#)

[Angular Reference](#)[jQuery Reference](#)

## Top Examples

[HTML Examples](#)[CSS Examples](#)[JavaScript Examples](#)[How To Examples](#)[SQL Examples](#)[Python Examples](#)[W3.CSS Examples](#)[Bootstrap Examples](#)[PHP Examples](#)[Java Examples](#)[XML Examples](#)[jQuery Examples](#)

## Web Courses

[HTML Course](#)[CSS Course](#)[JavaScript Course](#)[Front End Course](#)[SQL Course](#)[Python Course](#)[PHP Course](#)[jQuery Course](#)[Java Course](#)[C++ Course](#)[C# Course](#)[XML Course](#)[Get Certified »](#)

W3Schools is optimized for learning and training. Examples might be simplified to improve reading and learning. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using W3Schools, you agree to have read and accepted our [terms of use](#), [cookie and privacy policy](#).

Copyright 1999-2022 by Refsnes Data. All Rights Reserved.

W3Schools is Powered by W3.CSS.

