DATA

SCALING

A Numerical Dataset

object x_i	Height (H)	Weight (W)	Foot (F)	Label (L)
x_1	5.00	100	6	green
x_2	5.50	150	8	green
x_3	5.33	130	7	green
x_4	5.75	150	9	green
x_5	6.00	180	13	red
x_6	5.92	190	11	red
x_7	5.58	170	12	red
x_8	5.92	165	10	red

- $N = 8$ items
- $M = 3$ (unscaled) attributes

Code for the Dataset

```
import pandas as pd

data = pd.DataFrame(
 {"id":[ 1,2,3,4,5,6,7,8],
  "Label":["green","green","green","green",
                    "red","red","red","red"],
  "Height":[5,5.5,5.33,5.75,6.00,5.92,5.58,5.92],
  "Weight":[100,150,130,150,180,190,170,165],
  "Foot":[6, 8, 7, 9, 13, 11, 12, 10]},
  columns=["id","Height","Weight",
                          "Foot","Label"])
```

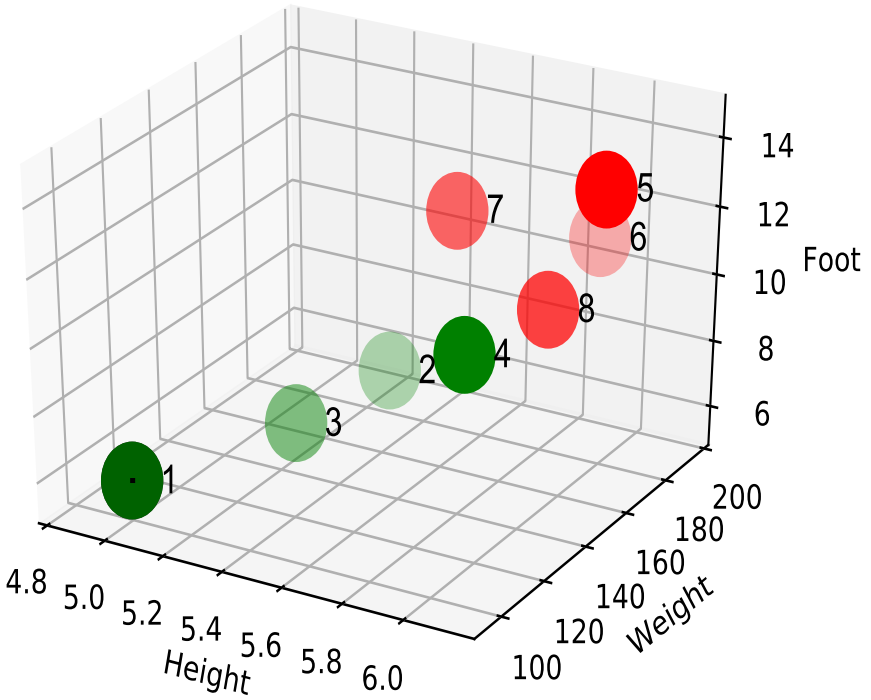
```
ipdb> data
  id Height Weight Foot Label
0  1   5.00    100    6  green
1  2   5.50    150    8  green
2  3   5.33    130    7  green
3  4   5.75    150    9  green
4  5   6.00    180   13    red
5  6   5.92    190   11    red
6  7   5.58    170   12    red
7  8   5.92    165   10    red
```

Need For Scaling

- features have different statistical distributions

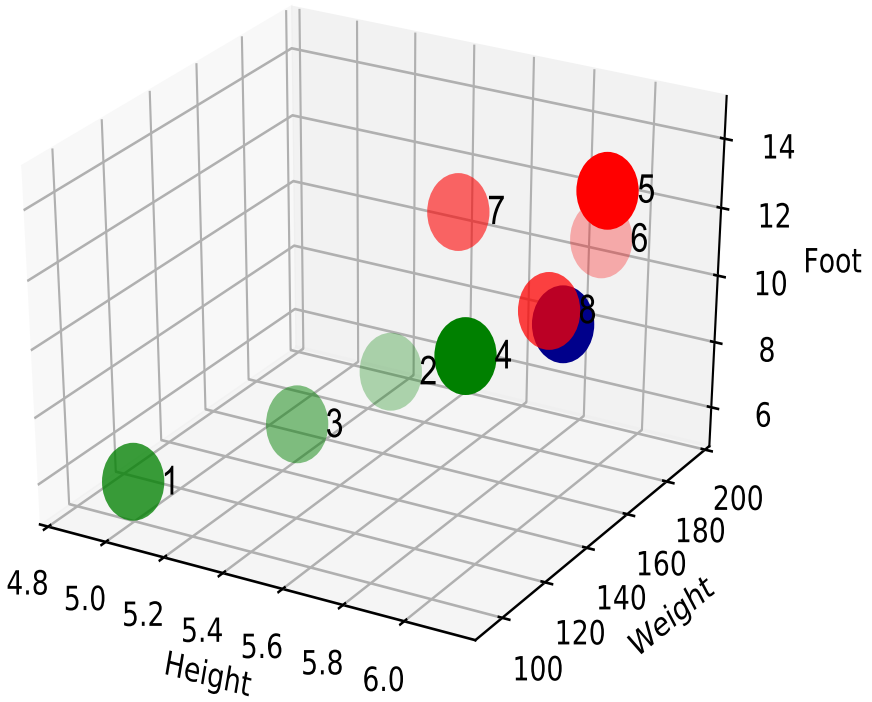
```
>> features=data[["eight", "Weight", "Foot"]]
>> features.describe()
          Height      Weight      Foot
count   8.000000    8.000000   8.00000
mean    5.625000  154.375000   9.50000
std     0.343428   28.962722   2.44949
min     5.000000  100.000000   6.00000
25%     5.457500  145.000000   7.75000
50%     5.665000  157.500000   9.50000
75%     5.920000  172.500000  11.25000
max     6.000000  190.000000  13.00000
```

A Dataset Illustration



- many methods use "distance"

A New Instance



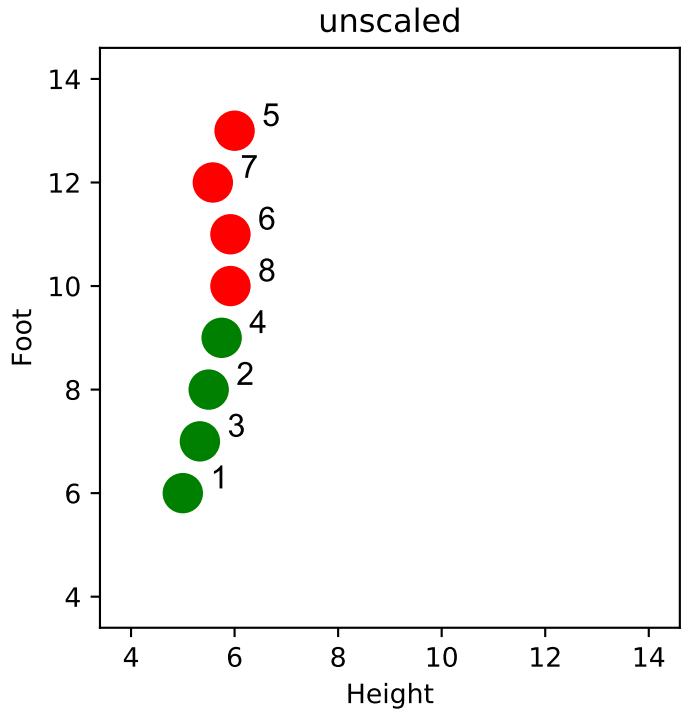
$$(\text{H}{=}6,\ \text{W}{=}160,\ \text{F}{=}10) \mapsto ?$$

No Scaling

```
import pandas as pd
data = pd.DataFrame(
 {"id":[ 1,2,3,4,5,6,7,8],
  "Label":["green","green","green","green",
                    "red","red","red","red"],
  "Height":[5,5.5,5.33,5.75,6.00,5.92,5.58,5.92],
  "Weight":[100,150,130,150,180,190,170,165],
  "Foot":[6, 8, 7, 9, 13, 11, 12, 10]},
  columns=["id","Height","Weight",
                           "Foot","Label"])
X = data[["Height", "Weight"]].values

ipdb> X
array([[  5.  ,   6.  ],
       [  5.5 ,   8.  ],
       [  5.33,   7.  ],
       [  5.75,   9.  ],
       [  6.  ,  13.  ],
       [  5.92,  11.  ],
       [  5.58,  12.  ],
       [  5.92,  10.  ]])
```

No Scaling




```
>> import numpy as np
>> np.linalg.norm(X[1,:] - X[2,:])
1.01434708064
>> np.linalg.norm(X[5,:] - X[6,:])
1.05621967412
```

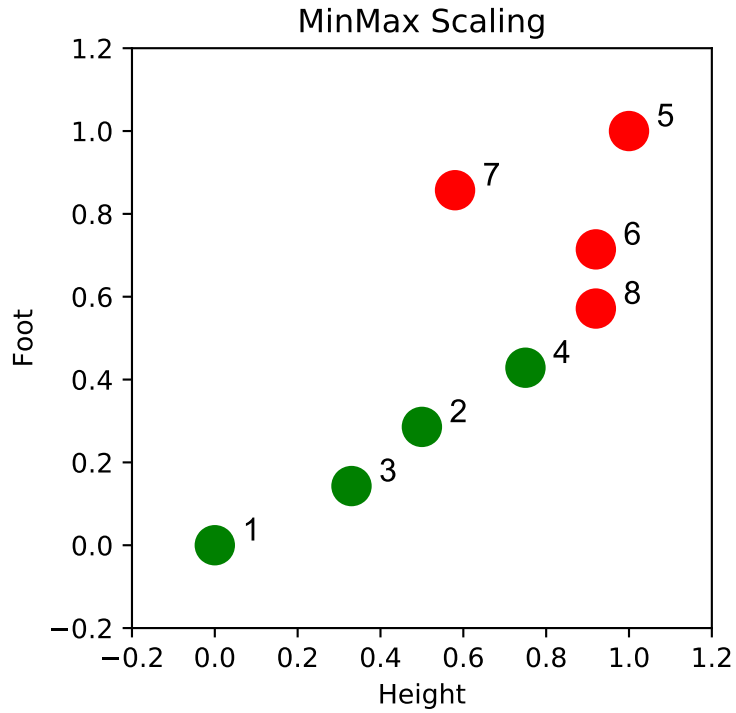
- unscaled $d\big(x_2, x_3\big) \approx d\big(x_6, x_7\big)$

Min-Max Scaling

```
import pandas as pd
data = pd.DataFrame(
 {"id":[ 1,2,3,4,5,6,7,8],
  "Label":["green","green","green","green",
                    "red","red","red","red"],
  "Height":[5,5.5,5.33,5.75,6.00,5.92,5.58,5.92],
  "Weight":[100,150,130,150,180,190,170,165],
  "Foot":[6, 8, 7, 9, 13, 11, 12, 10]},
  columns=["id","Height","Weight",
                           "Foot","Label"])
X = data[["Height", "Weight"]].values
Z = MinMaxScaler().fit_transform(X)

ipdb> Z
array([[ 0.        ,  0.        ],
       [ 0.5       ,  0.28571429],
       [ 0.33      ,  0.14285714],
       [ 0.75      ,  0.42857143],
       [ 1.        ,  1.        ],
       [ 0.92      ,  0.71428571],
       [ 0.58      ,  0.85714286],
       [ 0.92      ,  0.57142857]])
```

Min-Max Scaling




```
>> import numpy as np
>> np.linalg.norm(Z[1,:] - Z[2,:])
0.2220544151
>> np.linalg.norm(Z[5,:] - Z[6,:])
0.368792846006
```

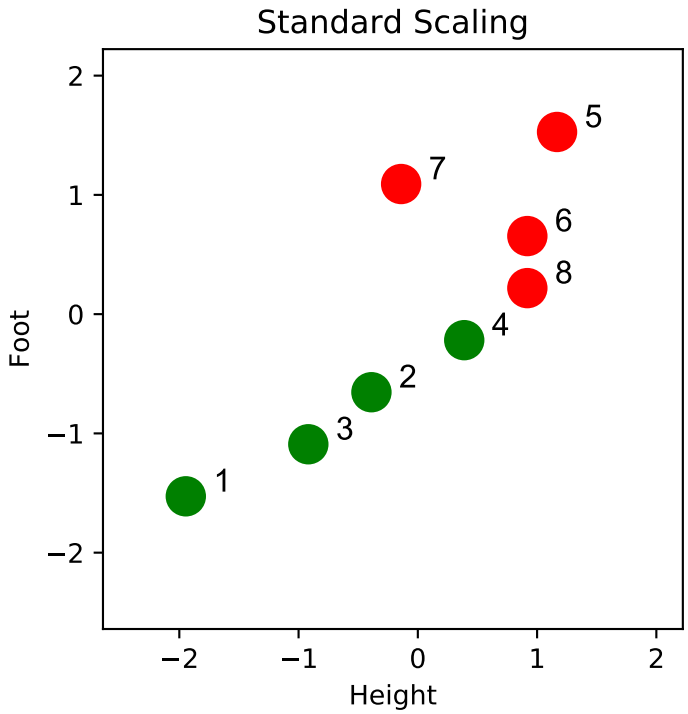
- min-max $d\left(x_2^*, x_3^*\right) < d\left(x_6^*, x_7^*\right)$

Standard Scaling

```
import pandas as pd
data = pd.DataFrame(
 {"id":[ 1,2,3,4,5,6,7,8],
  "Label":["green","green","green","green",
                    "red","red","red","red"],
  "Height":[5,5.5,5.33,5.75,6.00,5.92,5.58,5.92],
  "Weight":[100,150,130,150,180,190,170,165],
  "Foot":[6, 8, 7, 9, 13, 11, 12, 10]},
  columns=["id","Height","Weight",
                           "Foot","Label"])
X = data[["Height", "Weight"]].values
Z = StandardScaler().fit_transform(X)

ipdb> Z
array([[-1.94554002, -1.52752523],
       [-0.389108  , -0.65465367],
       [-0.91829489, -1.09108945],
       [ 0.389108  , -0.21821789],
       [ 1.16732401,  1.52752523],
       [ 0.91829489,  0.65465367],
       [-0.14007888,  1.09108945],
       [ 0.91829489,  0.21821789]])
```

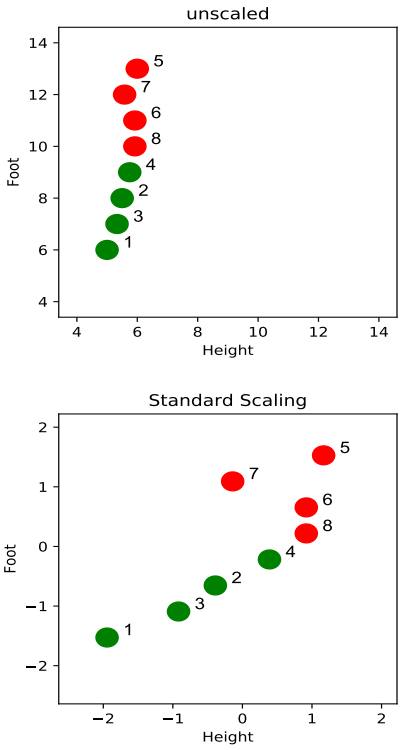
Standard Scaling




```
>> import numpy as np
>> np.linalg.norm(Z[1,:] - Z[2,:])
0.685940923233
>> np.linalg.norm(Z[5,:] - Z[6,:])
1.14482803479
```

- standard $d\left(x_2^{**}, x_3^{**}\right) < d\left(x_6^{**}, x_7^{**}\right)$

Effect of Scaling



- no scaling: $d(x_2, x_3) \approx d(x_6, x_7)$
- scaled: $d(x_2, x_3) > d(x_6, x_7)$

Concepts Check:

(a) need for scaling

(b) min-max scaling

(c) standard scaling