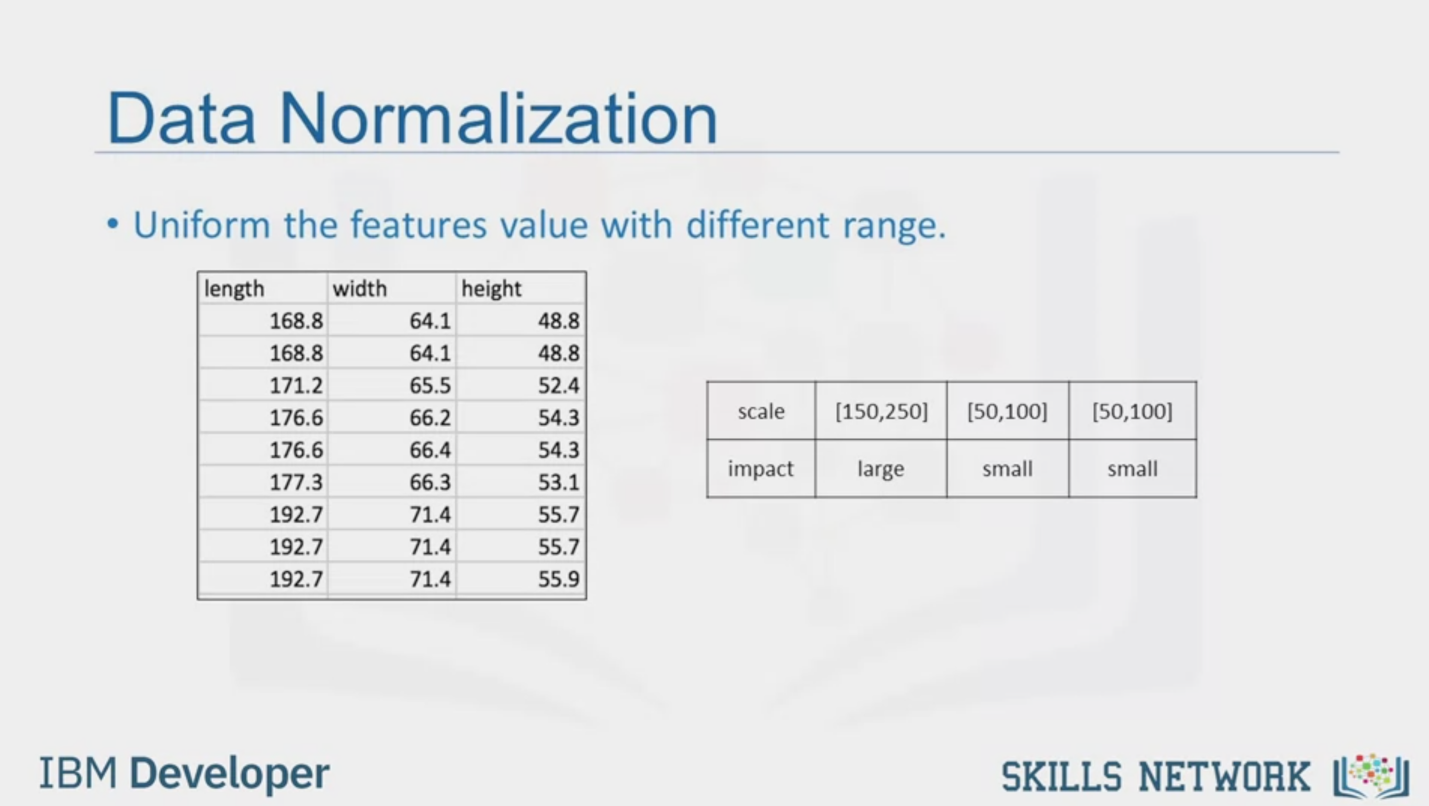


Data Normalization in Python

IBM Developer

SKILLS NETWORK



Data Normalization

• Uniform the features value with different range.

length

width

height

168.8

168.8

171.2

176.6

176.6

177.3

192.7

192.7

192.7

64.1

64.1

65.5

66.2

66.4

66.3

71.4

71.4

71.4

48.8

48.8

52.4

54.3

54.3

53.1

55.7

55.7

55.9

scale

impact

[150,250]

large

[50.1001

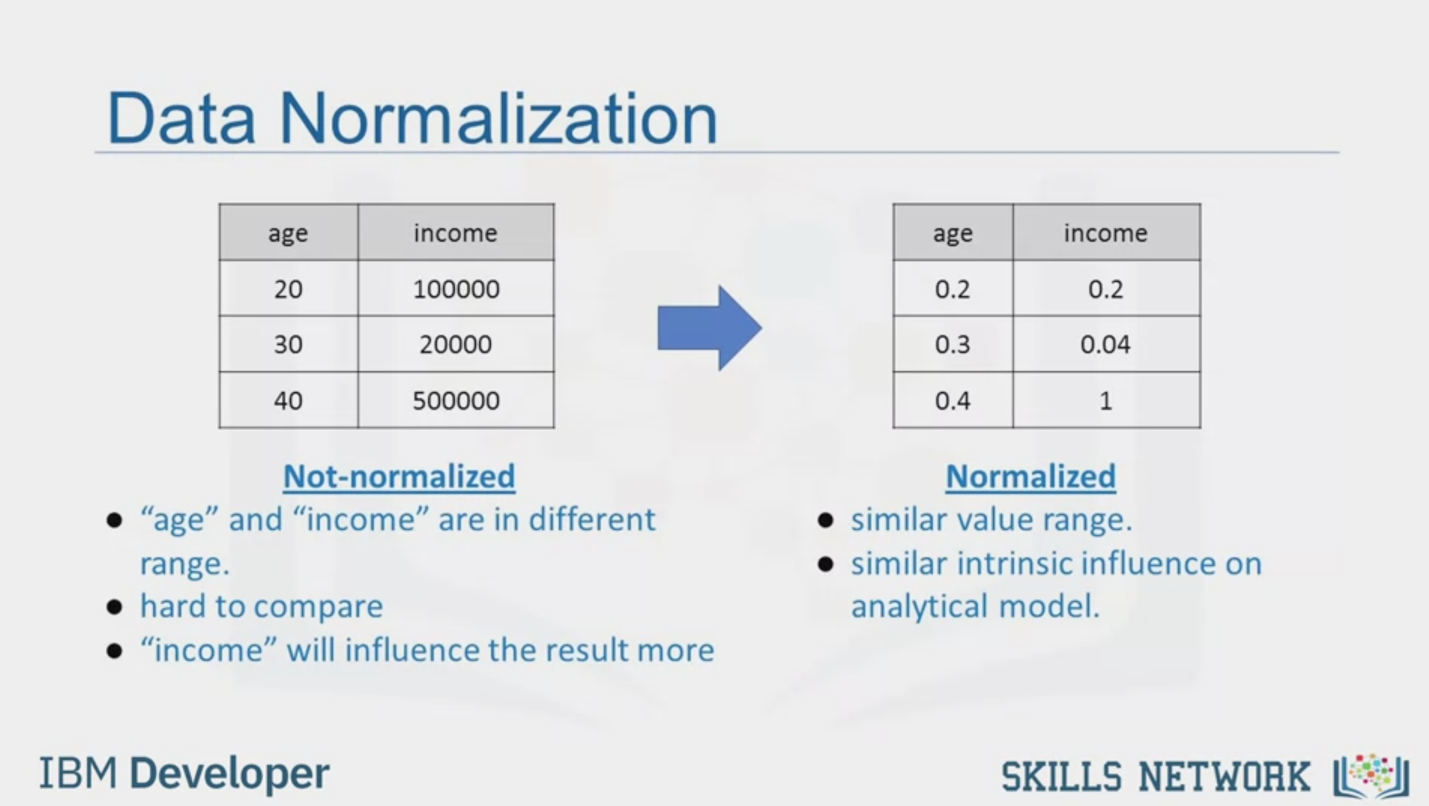
[50.1001

small

small

IBM Developer

SKILLS NETWORK



Data Normalization

age

20

30

40

income

100000

20000

500000

Not-normalized

"age" and "income" are in different

range.

hard to compare

"income" will influence the result more

age

0.2

0.3

0.4

income

0.2

0.04

1

Normalized

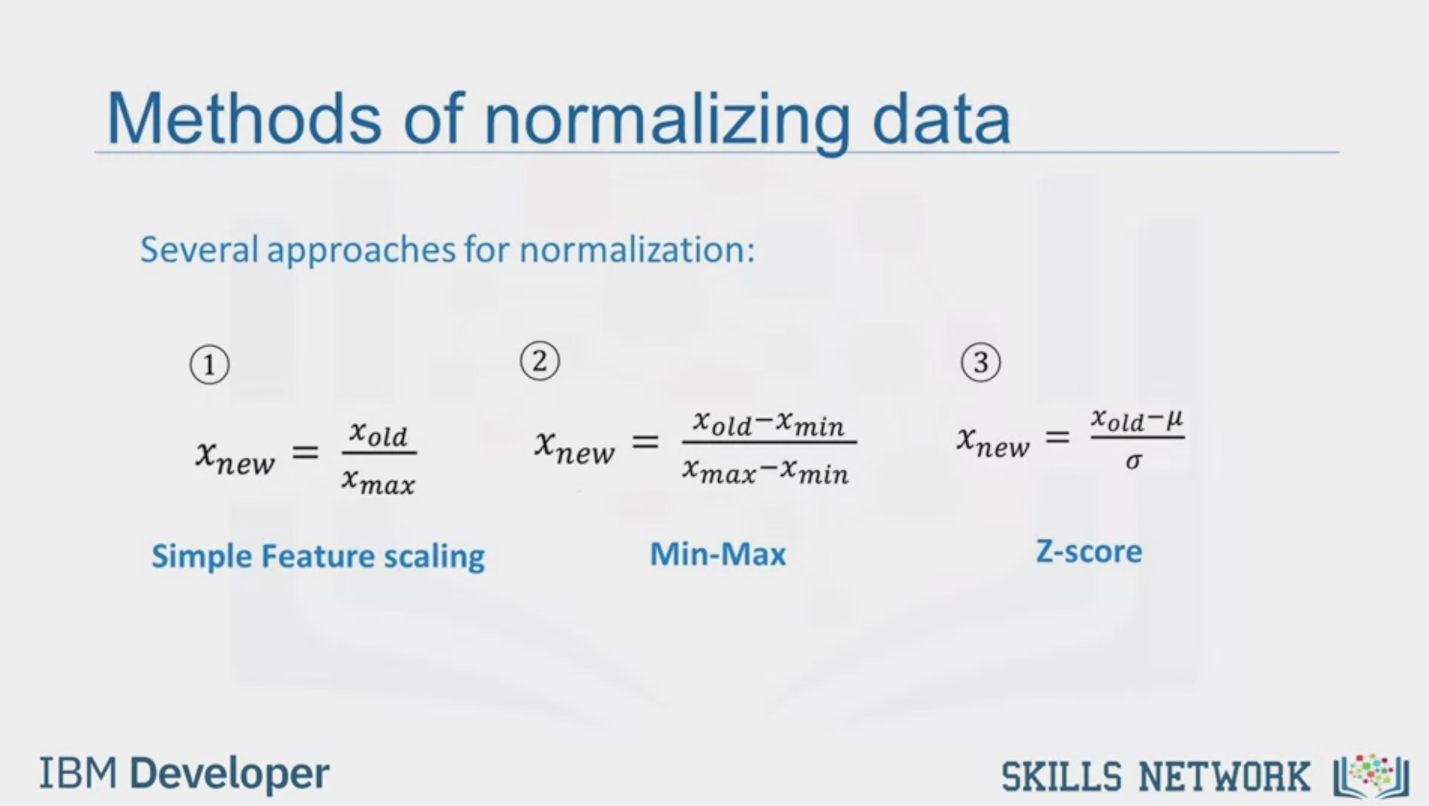
• similar value range.

• similar intrinsic influence on

analytical model.

IBM Developer

SKILLS NETWORK



Methods of normalizing data

Several approaches for normalization:

1

Xnew = Xold / Xmax

Simple Feature scaling

2

Xnew = Xold-Xmin / Xmax-Xmin

Min-Max

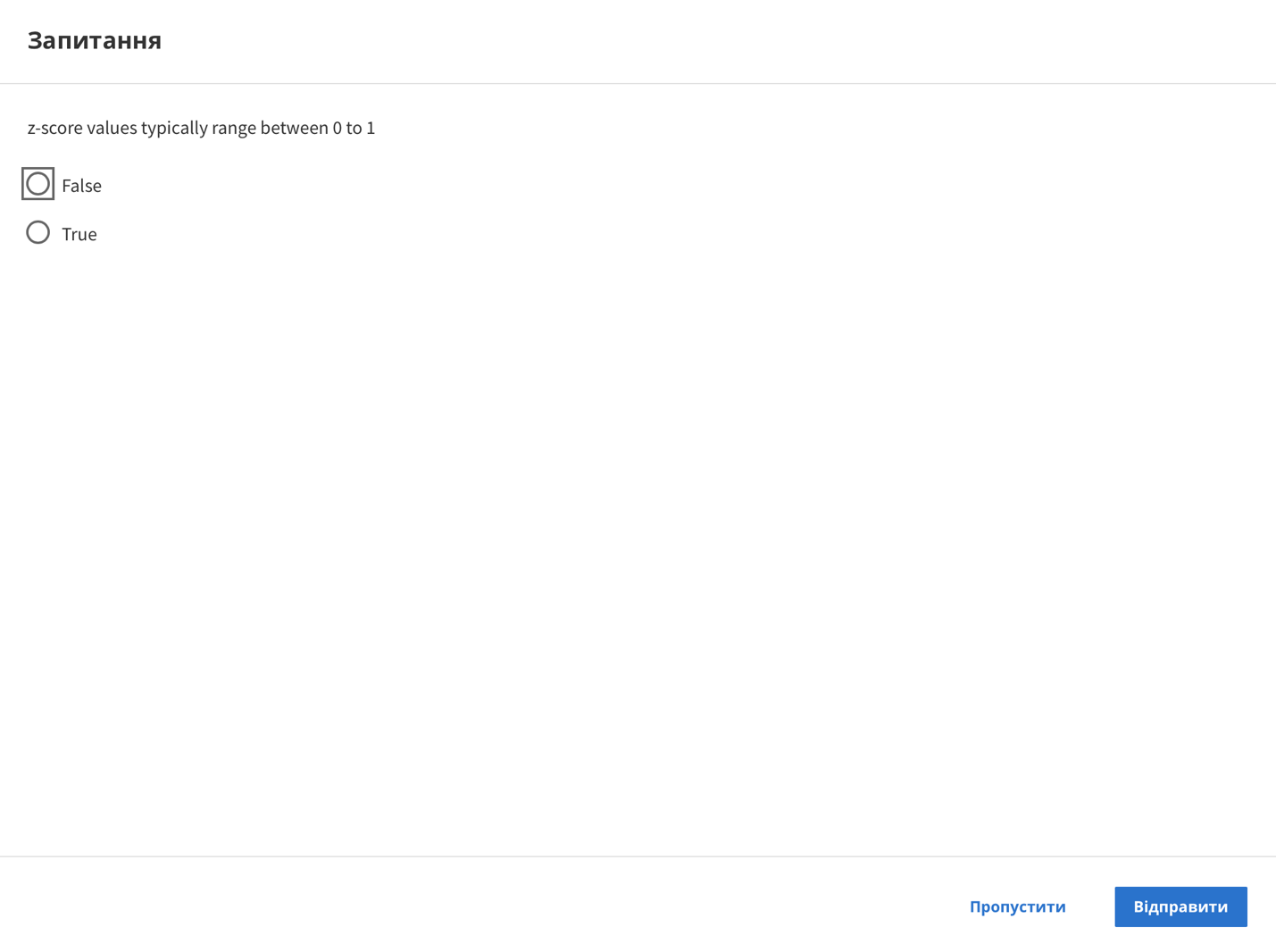
3

Xnew = Xold-μ/σ

Z-score

IBM Developer

SKILLS NETWORK



Question

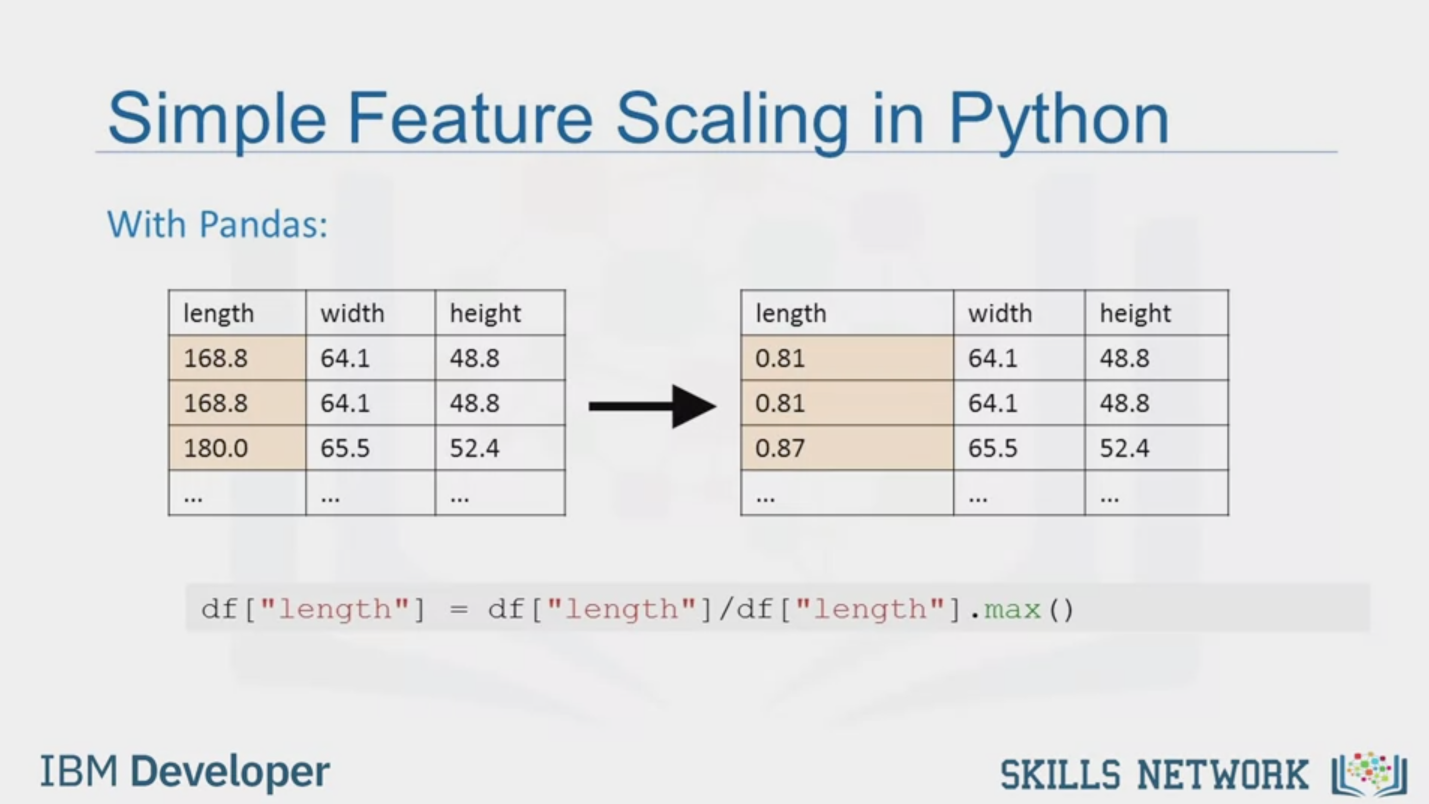
z-score values typically range between 0 to 1

False

True

Skip

Send



Simple Feature Scaling in Python

With Pandas:

length

168.8

168.8

180.0

width

64.1

64.1

65.5

height

48.8

48.8

52.4

length

0.81

0.81

0.87

width

64.1

64.1

65.5

height

48.8

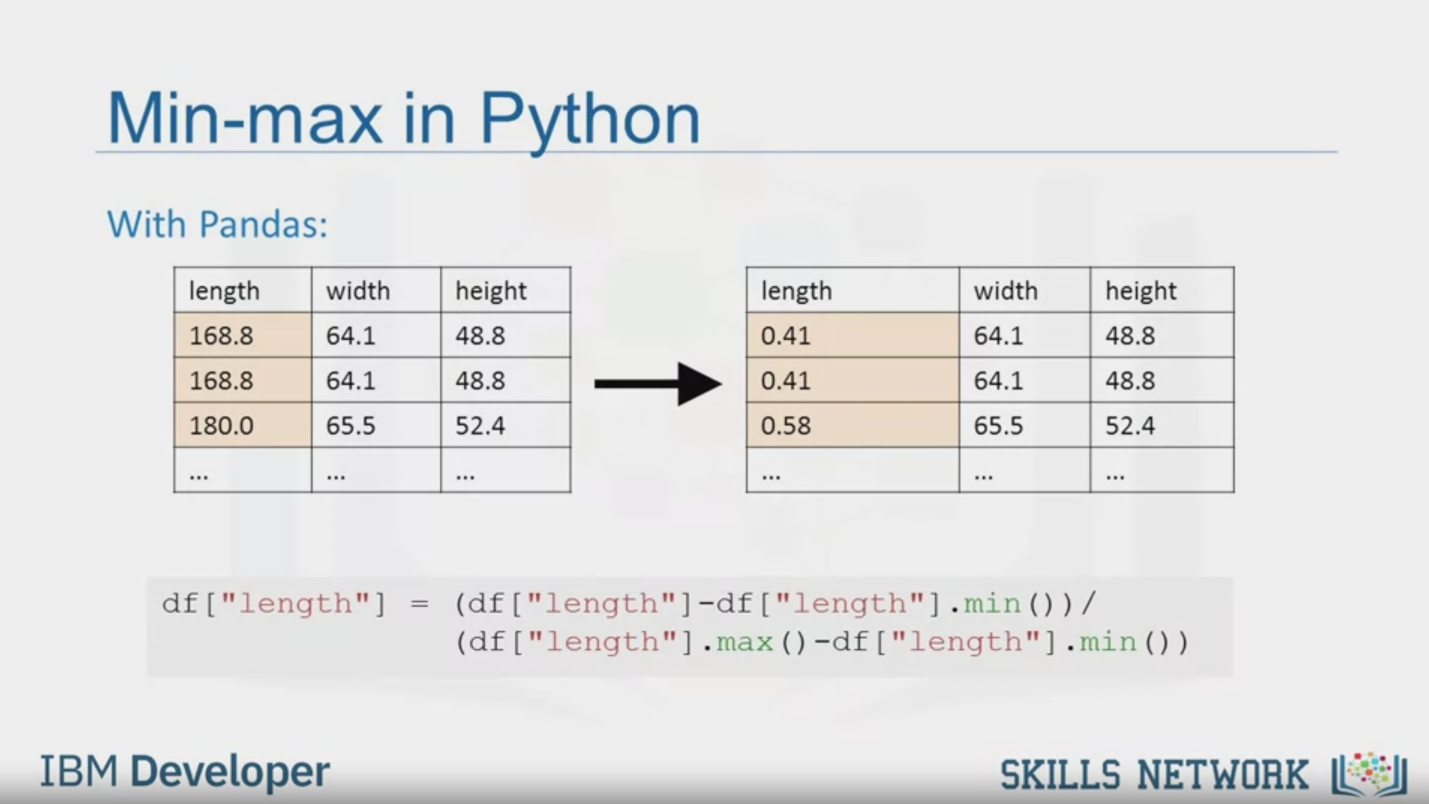
48.8

52.4

df ["length"] = df["length"]/df["length"].max ()

IBM Developer

SKILLS NETWORK



Min-max in Python

With Pandas:

length

168.8

168.8

180.0

width

64.1

64.1

65.5

height

48.8

48.8

52.4

length

0.41

0.41

0.58

width

64.1

64.1

65.5

height

48.8

48.8

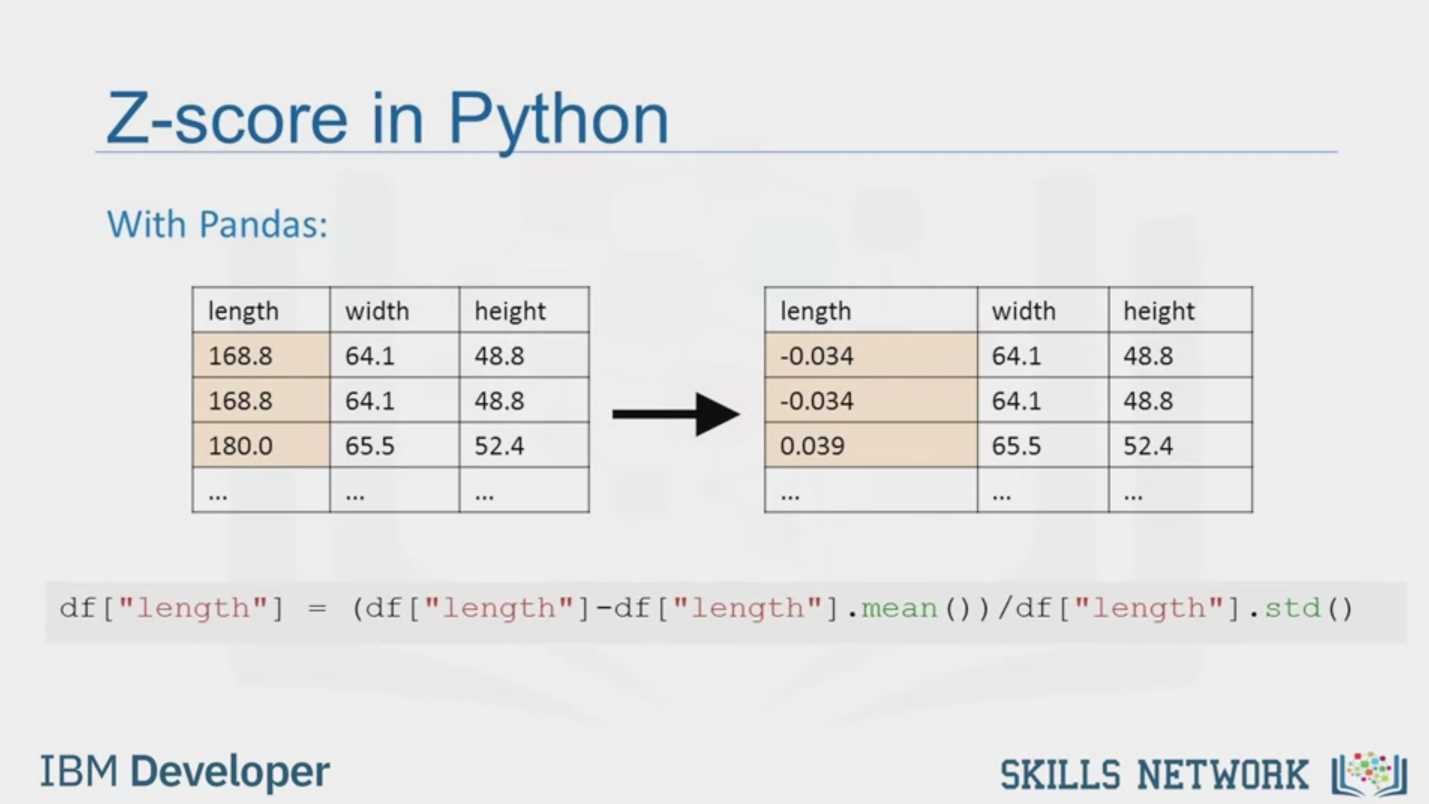
52.4

df["length"] = (df["length"] -df ["length"].min () ) /

(df["length"].max () - df ["length"].min () )

IBM Developer

SKILLS NETWORK



Z-score in Python

With Pandas:

length

168.8

168.8

180.0

width

64.1

64.1

65.5

height

48.8

48.8

52.4

length

-0.034

-0.034

0.039

width

641

64.1

65.5

height

48.8

48.8

52.4

df["length"] = (df ["length"]-df ["length"].mean () )/df["length"].std ()

IBM Developer

SKILLS NETWORK



SKILLS NETWORK

IBM Developer

In this video, we'll be talking about data normalization.

An important technique to understand in data pre-processing.

When we take a look at the used car data set,

we notice in the data that the feature length ranges from 150-250,

while feature width and height ranges from 50-100.

We may want to normalize these variables so that the range of the values is consistent.

This normalization can make some statistical analyses easier down the road.

By making the ranges consistent between variables,

normalization enables a fair comparison between the different features,

making sure they have the same impact.

It is also important for computational reasons.

Here is another example that will help you understand why normalization is important.

Consider a data set containing two features, age and income.

Where age ranges from 0-100,

while income ranges from 0-20,000 and higher.

Income is about 1,000 times larger than age and ranges from 20,000-500,000.

So, these two features are in very different ranges.

When we do further analysis,

like linear regression for example,

the attribute income will intrinsically

influence the result more due to its larger value.

But this doesn't necessarily mean it is more important as a predictor.

So, the nature of the data biases

the linear regression model to weigh income more heavily than age.

To avoid this, we can normalize

these two variables into values that range from zero to one.

Compare the two tables at the right.

After normalization, both variables now

have a similar influence on the models we will build later.

There are several ways to normalize data.

I will just outline three techniques.

The first method called simple feature scaling just

divides each value by the maximum value for that feature.

This makes the new values range between zero and one.

The second method called min-max takes

each value X\_old subtract it from the minimum value of that feature,

then divides by the range of that feature.

Again, the resulting new values range between zero and one.

The third method is called z-score or standard score.

In this formula for each value you subtract the mu which is the average of the feature,

and then divide by the standard deviation sigma.

The resulting values hover around zero,

and typically range between negative three and positive three but can be higher or lower.

Following our earlier example,

we can apply the normalization method on the length feature.

First, we use the simple feature scaling method,

where we divide it by the maximum value in the feature.

Using the pandas method max,

this can be done in just one line of code.

Here's the min-max method on the length feature.

We subtract each value by the minimum of that column,

then divide it by the range of that column.

The max minus the min.

Finally, we apply the z-score method on length feature to normalize the values.

Here we apply the mean and STD method on the length feature.

Mean method will return the average value of the feature in the data set,

and STD method will return the standard deviation of the features in the data set.