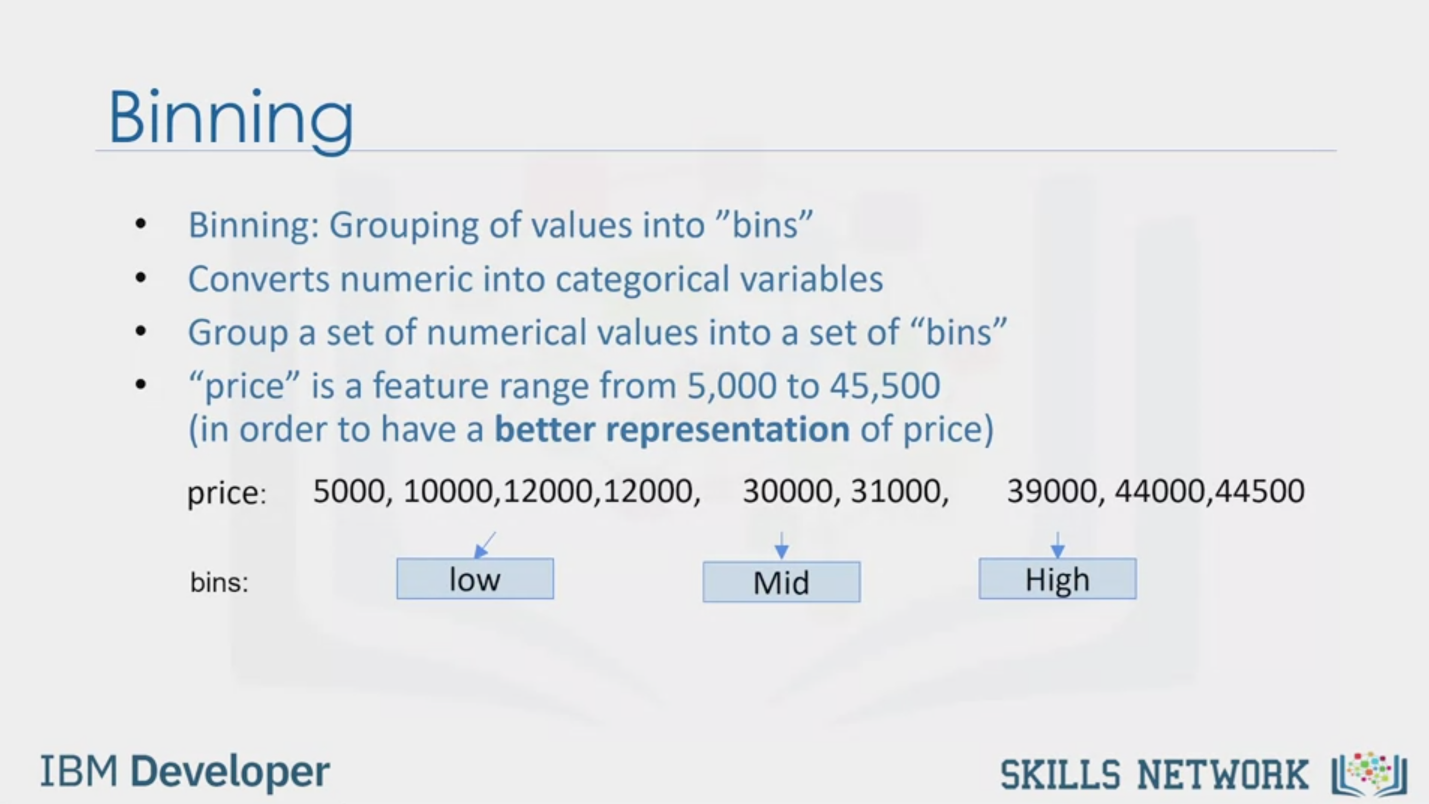


Binning in Python

IBM Developer

SKILLS NETWORK



Binning

Binning: Grouping of values into "bins"

Converts numeric into categorical variables

Group a set of numerical values into a set of "bins"

"price" is a feature range from 5,000 to 45,500

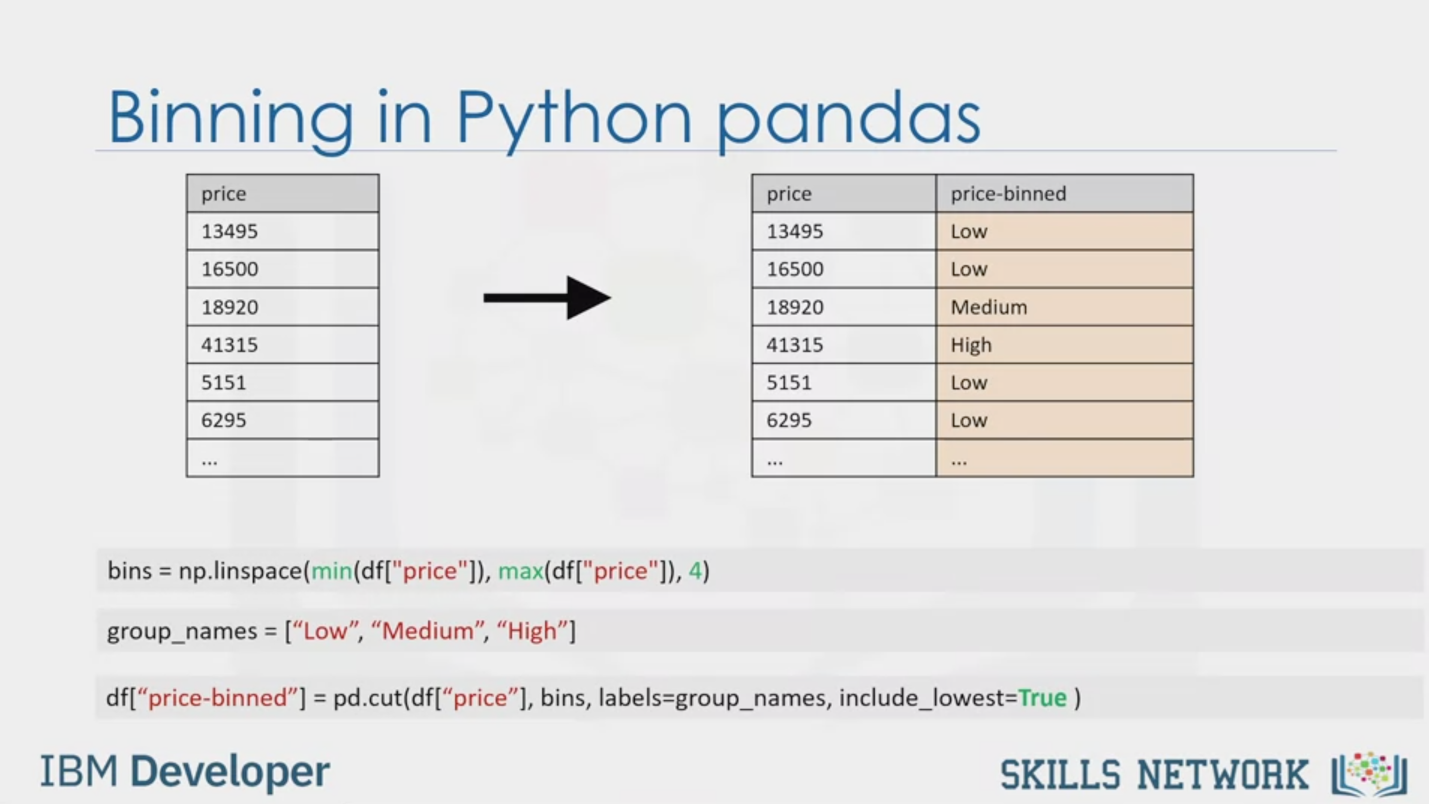
(in order to have a better representation of price)

price: 5000, 10000,12000,12000, 30000, 31000, 39000, 44000,44500

bins: low Mid High

IBM Developer

SKILLS NETWORK



Binning in Python pandas

price

13495

16500

18920

41315

5151

6295

price

13495

16500

18920

41315

5151

6295

price-binned

Low

LOW

Medium

High

LOW

LOW

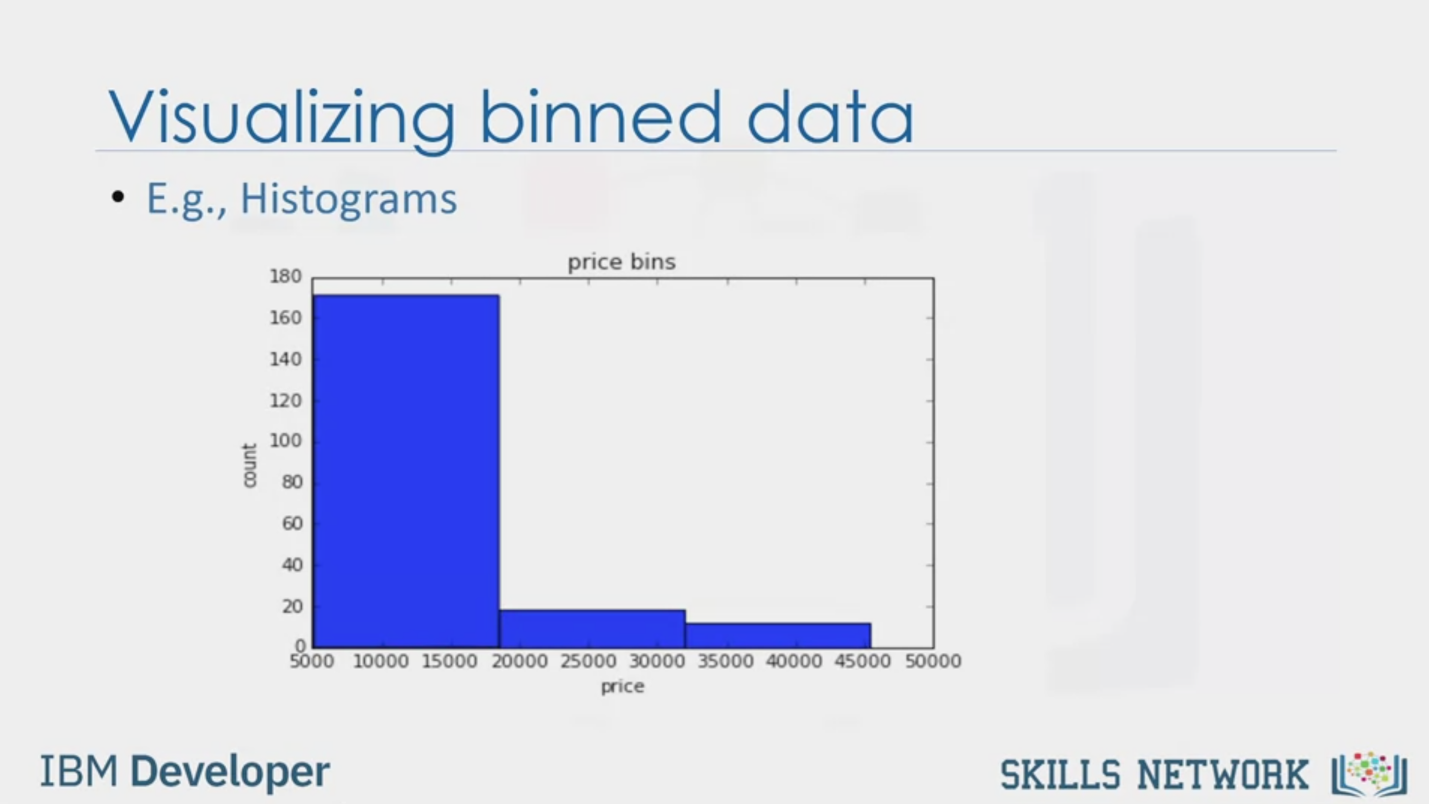
bins = np.linspace(min(df["price"]), max(df["price"]), 4)

group\_names = ["Low", "Medium", "High"]

df["price-binned"] = p.cut(df["price"], bins, labels=group\_names, include\_lowest=True )

IBM Developer

SKILLS NETWORK



Visualizing binned data

• E.g., Histograms

price bins

180

160

140

count

100

80

60

40

20

5000 10000 15000 20000 25000 30000 35000 40000 45000 50000

price

IBM Developer

SKILLS NETWORK



SKILLS NETWORK

IBM Developer

In this video, we’ll talk about binning as a method of data pre-processing.

Binning is when you group values together into bins. For example, you can bin “age”

into [0 to 5], [6 to 10], [11 to 15] and so on.

Sometimes, binning can improve accuracy of the predictive models.

In addition, sometimes we use data binning to group a set of numerical values into a

smaller number of bins to have a better understanding of the data distribution.

As example, “price” here is an attribute range from 5,000 to 45,500.

Using binning, we categorize the price into three bins: low price, medium price, and high

prices.

In the actual car dataset, ”price" is a numerical variable ranging from 5188 to 45400,

it has 201 unique values.

We can categorize them into 3 bins: low, medium, and high-priced cars.

In Python we can easily implement the binning: We would like 3 bins of equal binwidth, so

we need 4 numbers as dividers that are equal distance apart.

First we use the numpy function “linspace” to return the array “bins” that contains

4 equally spaced numbers over the specified interval of the price.

We create a list “group\_names “ that contains the different bin names.

We use the pandas function ”cut” to segment and sort the data values into bins.

You can then use histograms to visualize the distribution of the data after they’ve been

divided into bins.

This is the histogram that we plotted based on the binning that we applied in the price

feature.

From the plot, it is clear that most cars have a low price, and only very few cars have

high price.