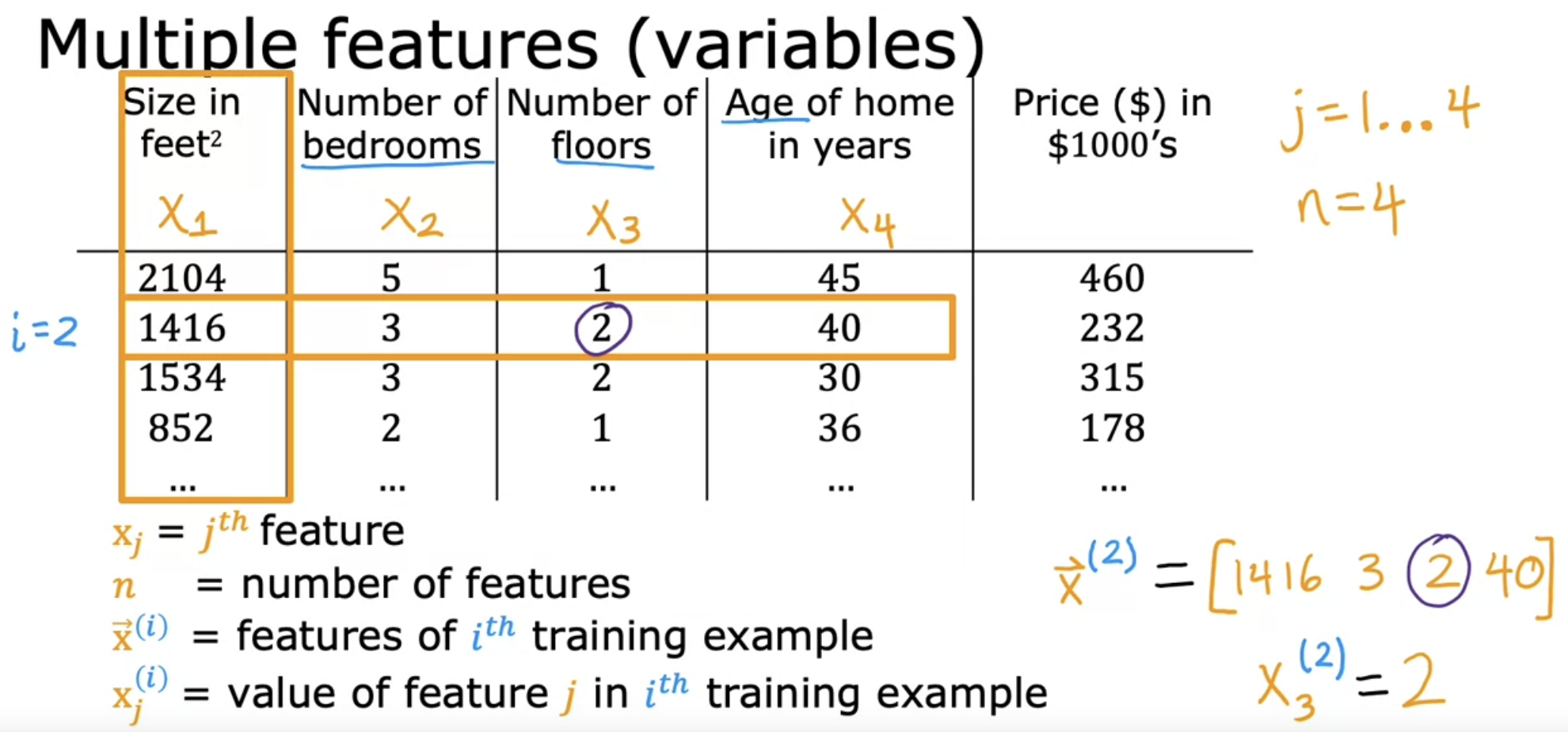
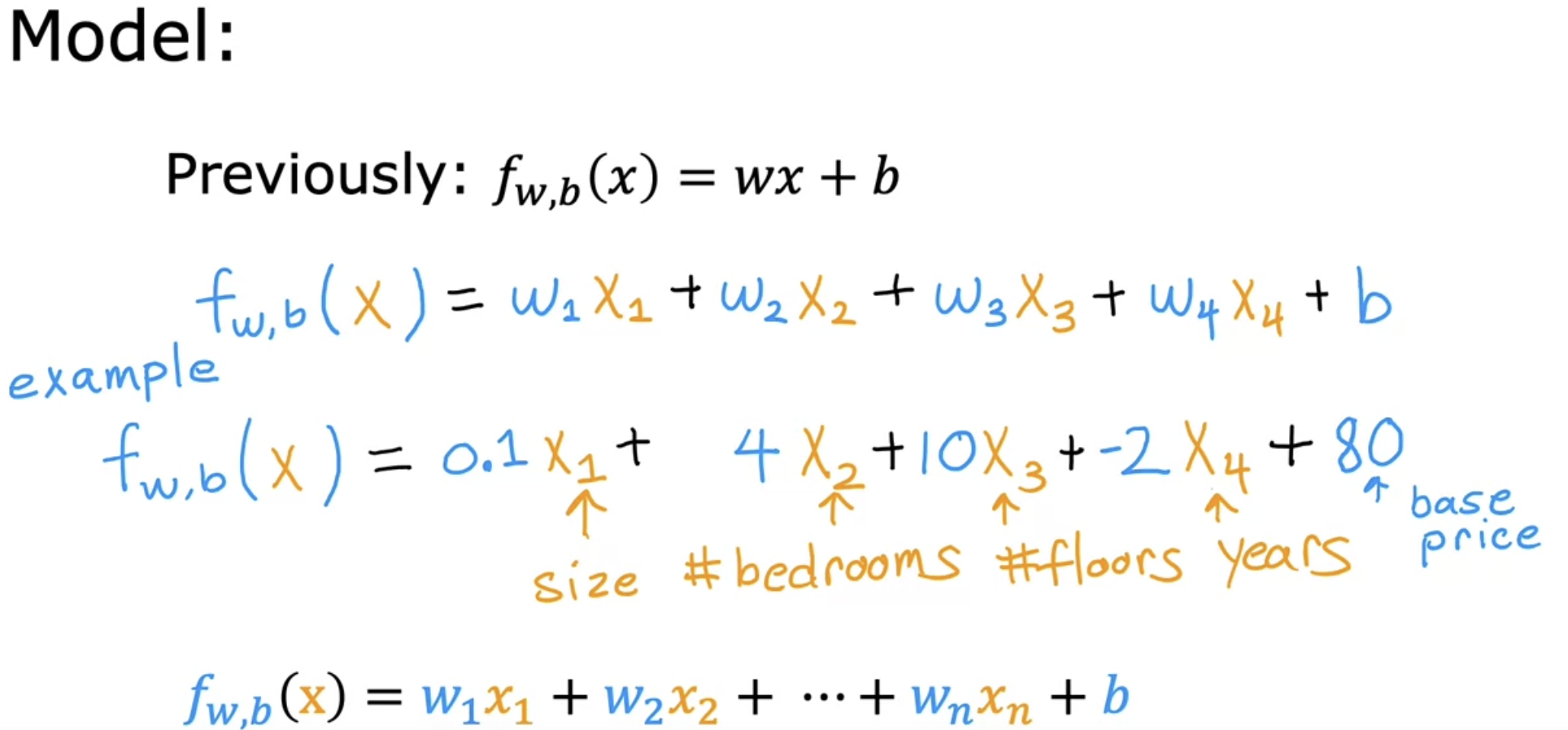
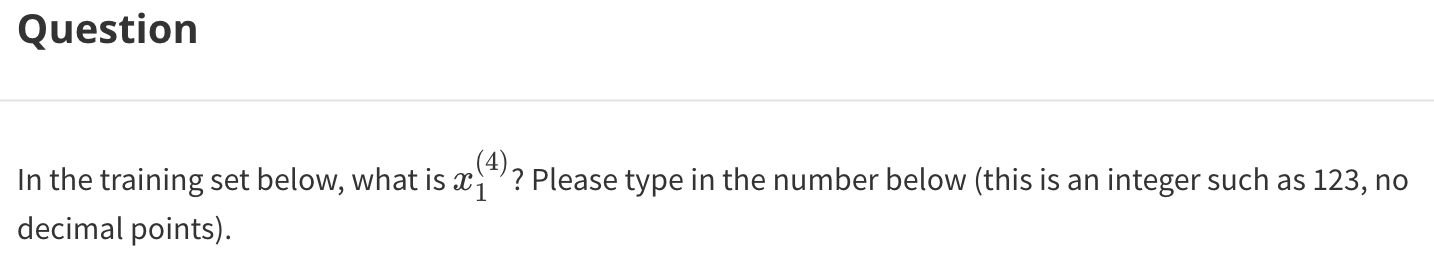
# **Multiple features**



3rd feature and 2nd training example



  
A picture containing table

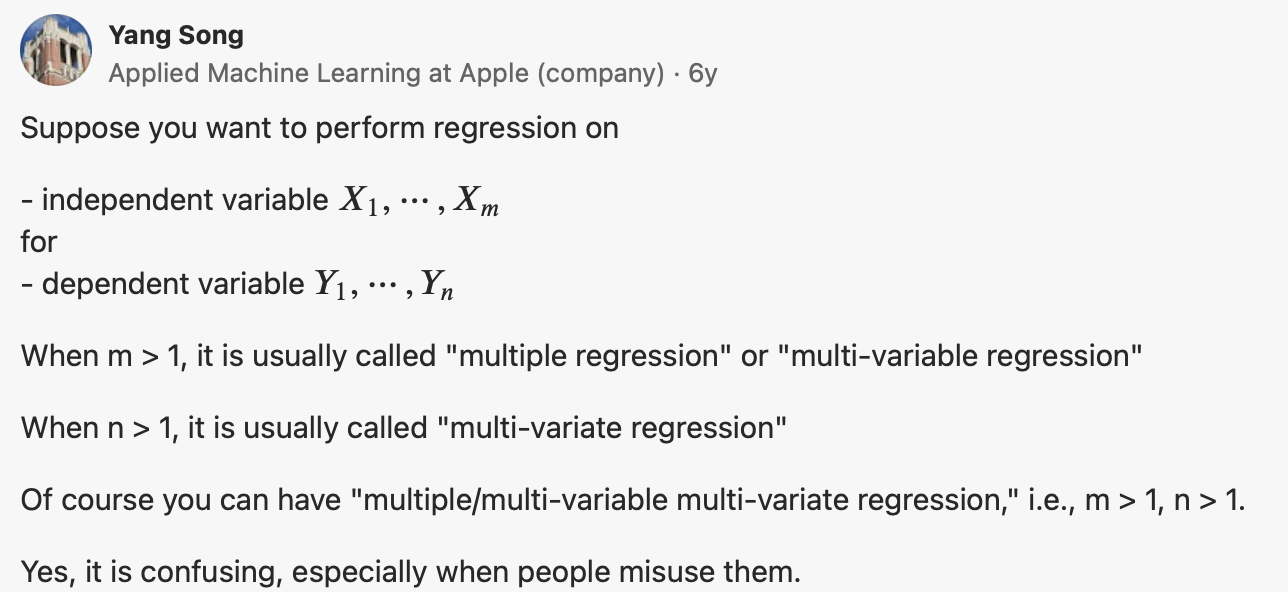
Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Difference between multiple regression and multi-variate regression.



# **Vectorization part-1**

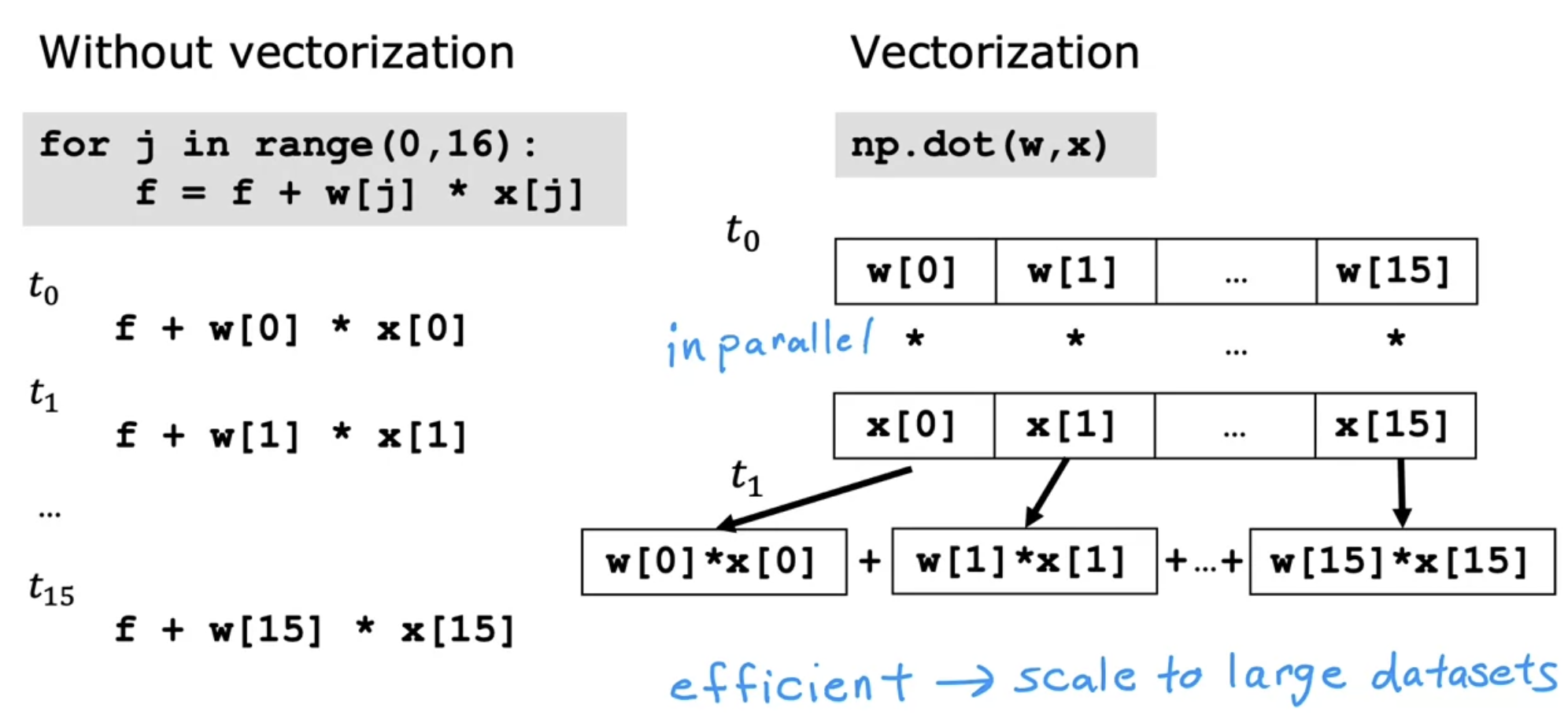
A picture containing text

Description automatically generated

Graphical user interface, text, application, email

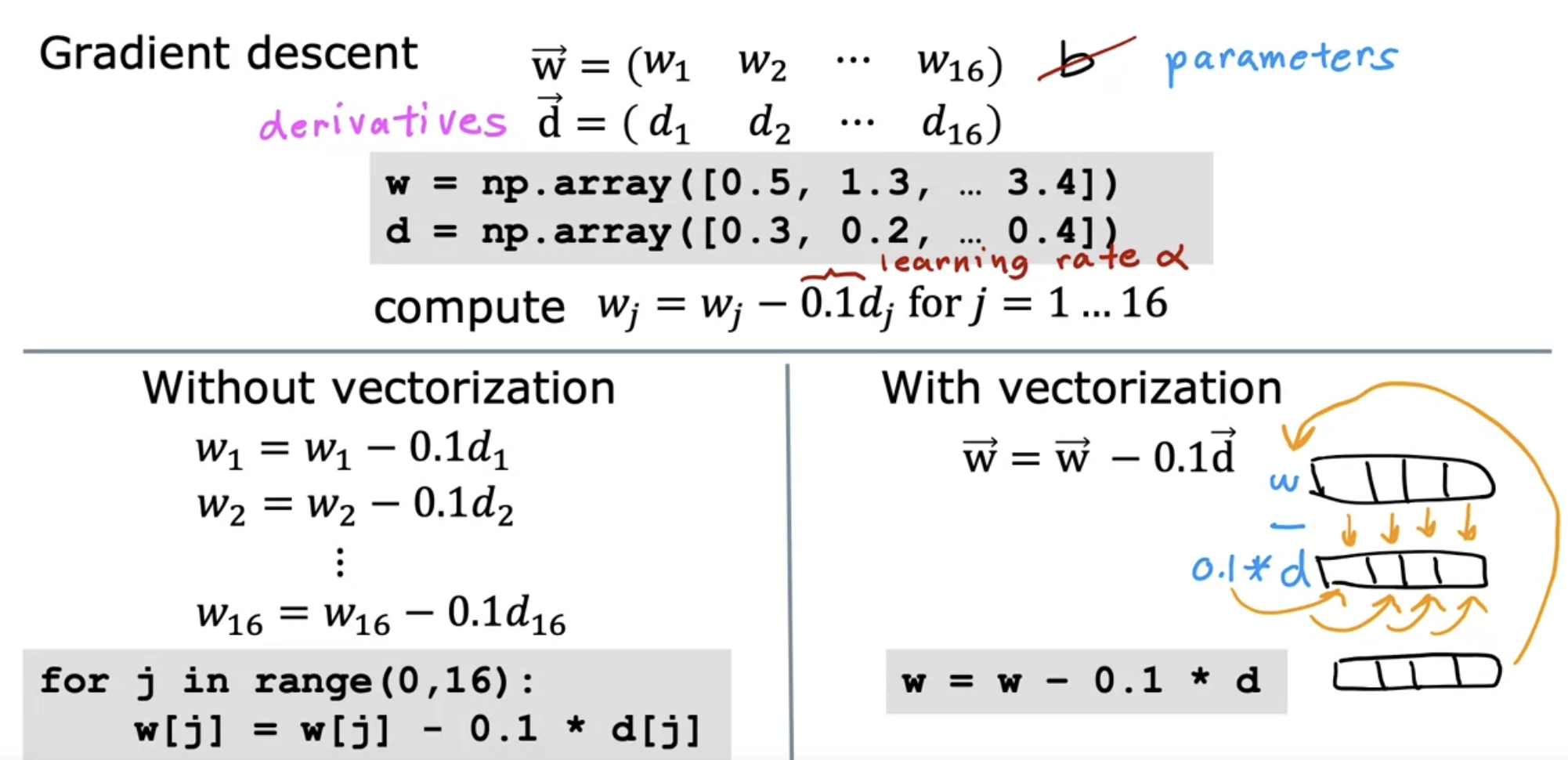
Description automatically generated

# **Vectorization part-2**



The computer can get all values of the vectors w and x, and in a single-step, it multiplies each pair of w and x with each other all at the same time in parallel.

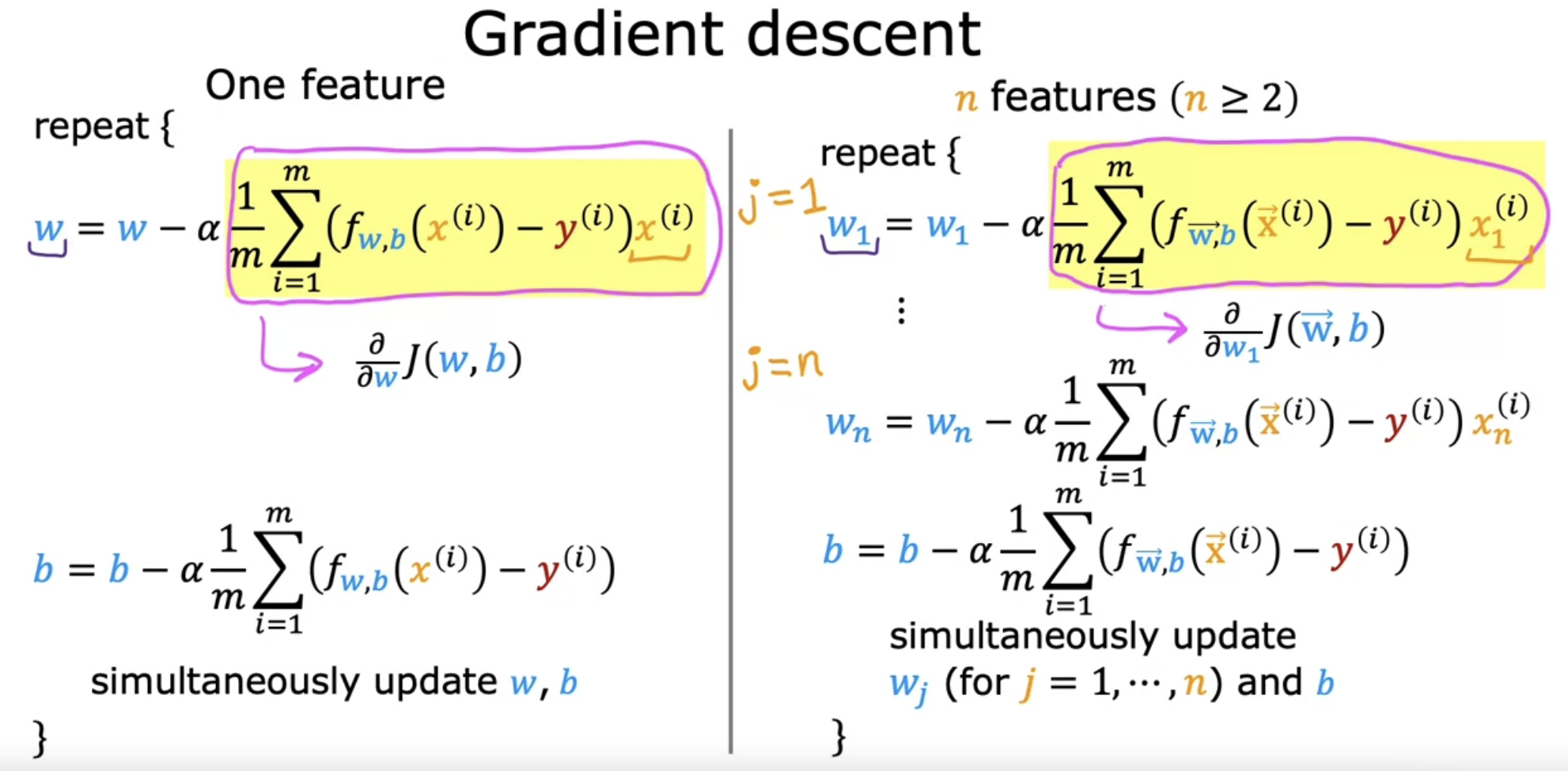
Then after that, the computer takes these 16 numbers and uses specialized hardware to add them altogether very efficiently, rather than needing to carry out distinct additions one after another to add up these 16 numbers.



In contrast, with factorization, you can imagine the computer's parallel processing hardware like this. It takes all 16 values in the vector w and subtracts in parallel,   
0.1 times all 16 values in the vector d, and assign all 16 calculations back to w   
all at the same time and all in one step. In code, you can implement this as follows, w is assigned to w minus 0.1 times d.

Refer 🡪 1\_Python\_Numpy\_Vectorization\_Solution.ipynb

# **Gradient Descent for multiple linear regression**

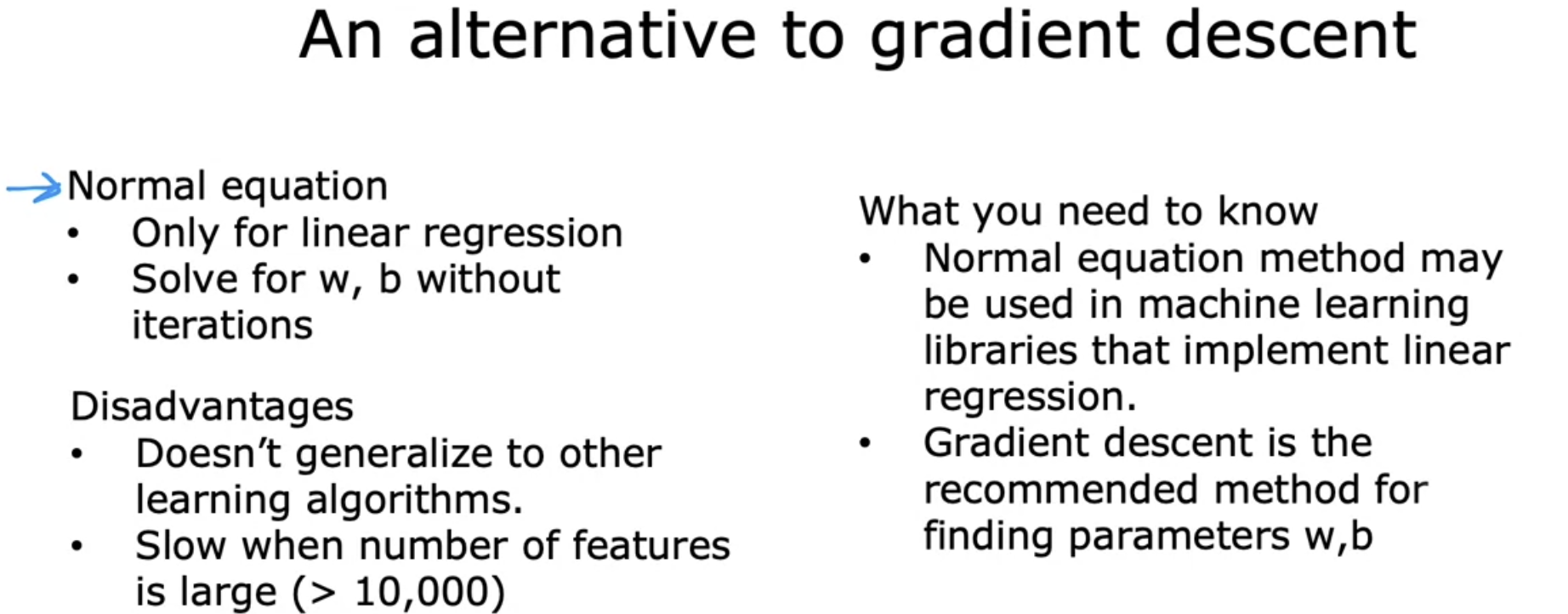


j=1 🡪 x1 ^ (i) 🡪 1st feature  
j=2 🡪 x2 ^ (i) 🡪 2nd feature

.

.

j=n 🡪 xn ^ (i) 🡪 nth feature



# **Multiple linear regression**

# **References**

<https://www.quora.com/What-is-the-difference-between-a-multiple-linear-regression-and-a-multivariate-regression>