Announcements

- Quiz 0 will be out on Canvas tonight at 5pm. Honorlock Based. Open Notes. Can attempt anywhere
- This is basically a syllabus quiz. It does not have questions related to the lecture content.
- This quiz is multiple attempts. Due at the end of the semester
- Quiz 1 onwards will cover lecture content, will be timed, will be 1 attempt only and will be due every week
- Lectures are recorded for you to revisit later. You can find it under Media Gallery.
- Project Pinned Post on Ed discussion for you to look for teammates

Georgia Tech

Georgia Tech

Machine Learning CS 4641 B

Data Analysis Toolbox 1

Nimisha Roy

Lecturer, College of Computing

Director, Online Undergraduate Initiatives

Georgia Tech

Setting up

- **Jupyter Notebooks (.ipynb)**
  Combine code, text, and visualizations in one interactive document.

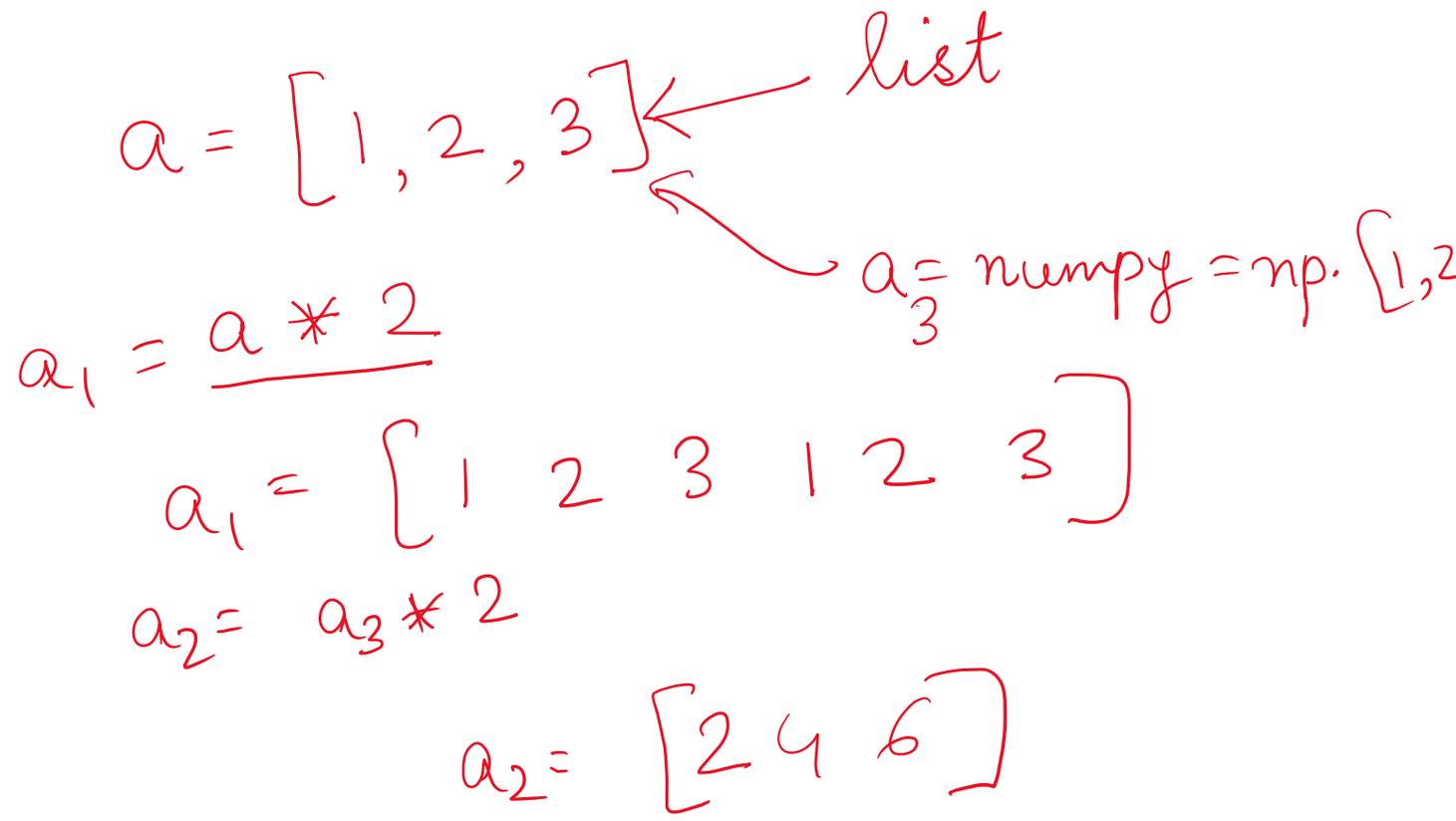
- **Option 1: Anaconda (Recommended for beginners)**
- Install Anaconda to get Python, Jupyter Notebook, and NumPy pre-packaged.
- Great for working offline on your computer.

- **Option 2: Google Colab**
- Free, browser-based environment with Jupyter-like notebooks.
- No installation needed — runs on Google's cloud.
- Easy to share and collaborate with classmates.

Georgia Tech

Python List vs NumPy Array

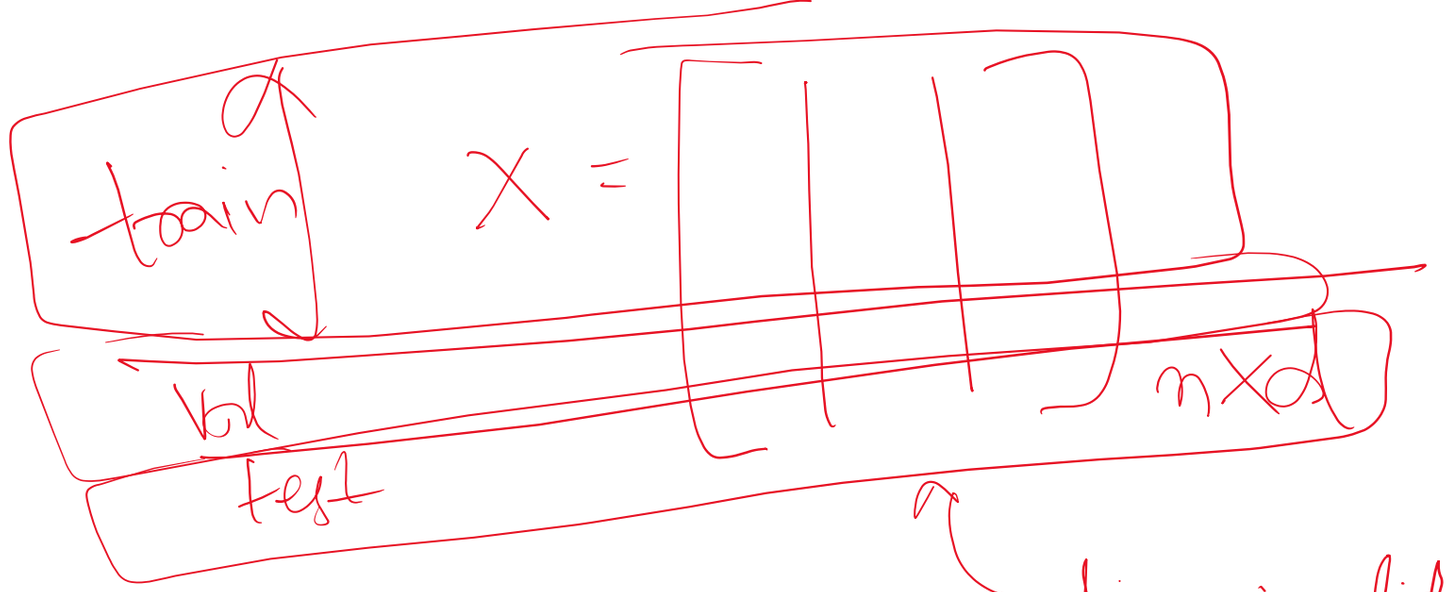
Python Lists
- Can store mixed data types
- No element-wise math
- Slower for numerical tasks



NumPy Arrays
- **Size** – more compact in memory
- **Performance** – optimized C backend, much faster
- **Functionality** – built-in math, linear algebra, and SciPy support
- Homogeneous data → efficient computations
- Vectorized operations (a+b, a*b)

👉 Takeaway: NumPy arrays are the core data structure for efficient, large-scale numerical computing in Python.

Georgia Tech



dimensionality reduction

X

data → n x d ← features

datapoints

We can splice rows → if we split data into train, val & test sets

We can splice columns → if we need to reduce dimensions (dimensionality reduction) among others

Georgia Tech

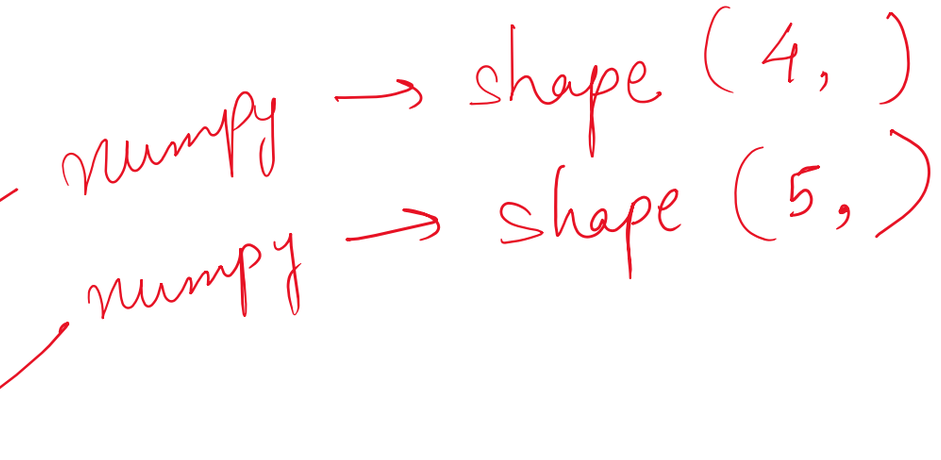
NumPy Broadcasting

- Perform operations on arrays of **different shapes**
- NumPy automatically expands dimensions when possible
- Eliminates the need for manual loops or reshaping
- Makes code **shorter, faster, and memory-efficient**

A = [1, 2, 3, 4]

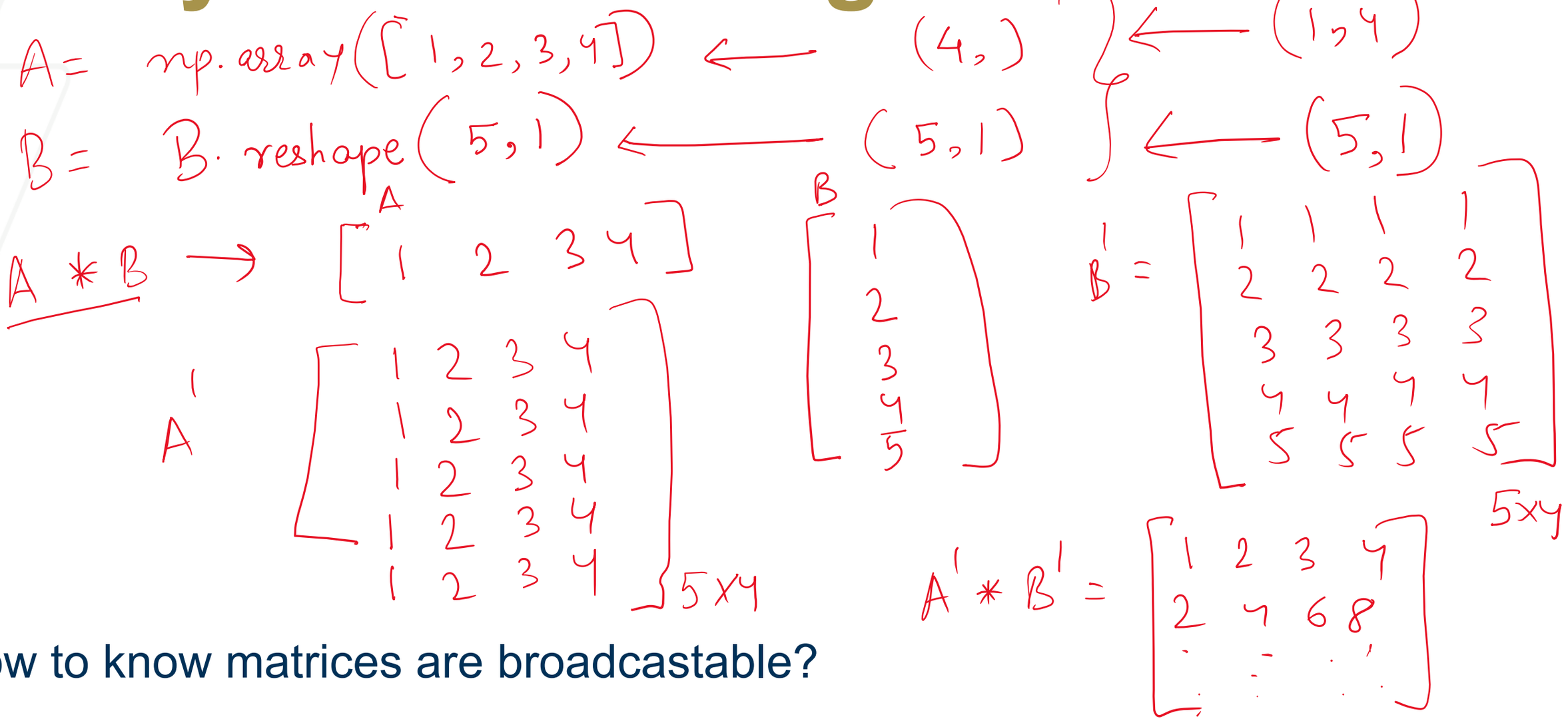
B = [1, 2, 3,4 ,5]

Is A*B possible?



Georgia Tech

NumPy Broadcasting

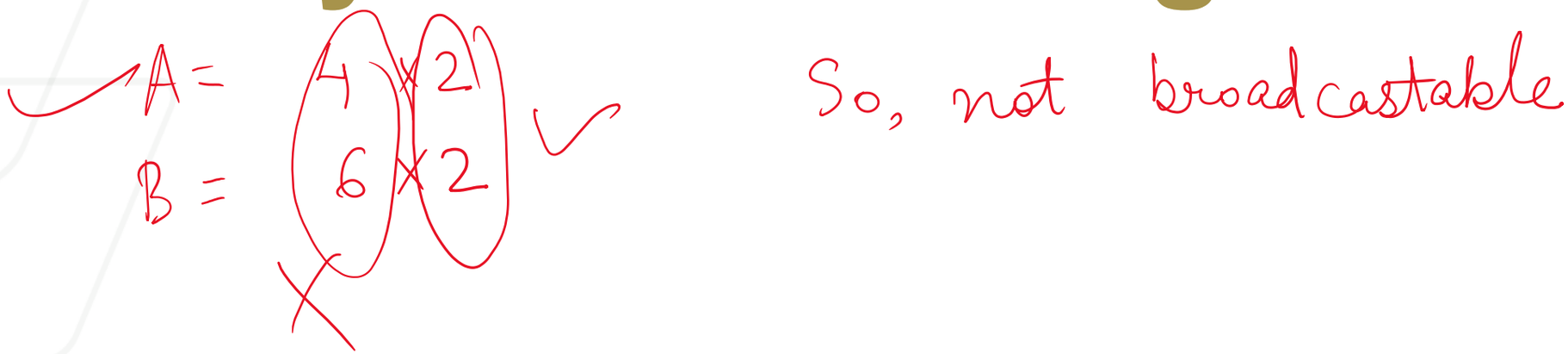


How to know matrices are broadcastable?

When two arrays don't have the same number of dimensions:
NumPy **prepends 1's to the smaller shape** until both arrays have the same number of dimensions.
Then it compares dimensions from **right → left**.

Georgia Tech

NumPy Broadcasting



👉 **Takeaway:** Broadcasting allows NumPy to apply operations across arrays of different sizes seamlessly

Much faster than for loops

Summary

- **Lists vs Arrays**
  - Arrays are smaller in memory, faster, and packed with math functions
- **Vectorization**
  - Write math like math: a + b instead of looping
  - Faster because operations run in optimized C code
- **Broadcasting**
  - Extends operations to arrays of different shapes automatically
  - No need for manual resizing or nested loops

👉 **Key Takeaway:**

With NumPy, we **don’t like for loops** — arrays, vectorization, and broadcasting make code cleaner, faster, and easier to reason about.

Georgia Tech