Why do we have 1D, 2D or 3D arrays and where do we need them in Data Science?

1D, 2D, and 3D arrays are essential data structures in programming and are commonly used in data science. Each type of array has its own specific use case.

1D Arrays: 1D arrays store elements of the same data type, such as numbers or strings. They are also known as lists or vectors.

Use Cases:
When you want to store a series of data, such as a list of numbers or a list of names.
When you want to apply an operation on all the elements of an array.

2D Arrays: 2D arrays store elements of the same data type arranged in a matrix format. They are also known as tables or grids.

Use Cases:
When you want to store data in a tabular format, such as a table of values or a matrix of images.
When you want to perform operations on rows or columns of an array.

3D Arrays: 3D arrays store elements of the same data type arranged in a cube format. They are also known as tensors.

Use Cases:
When you want to store data in a cube-like format, such as a cube of values or a tensor of images.
When you want to perform operations on layers or planes of an array.

In Data Science, arrays are commonly used in the following scenarios:
When dealing with data from databases, APIs, or data files.
When performing operations on individual elements of an array.
When reshaping arrays for compatibility with other functions or libraries.
When dealing with multi-dimensional data, such as images, sound, or video.

By understanding the characteristics and use cases of 1D, 2D, and 3D arrays, data scientists can make informed decisions about the best data structures for their specific needs.