PRESENTATION ABOUT NUMPY

Made by : Rawan Hatem

INTRODUCTION

- Introduction to NumPy for Machine Learning
  - NumPy is a special type of array in Python that's more powerful and efficient than regular lists
- Basic Python Lists vs. NumPy Arrays
  - Python lists can have different data types, which can be slow and computationally expensive for large datasets
  - NumPy arrays require all elements to be of the same data type, making them faster and more efficient
- Creating NumPy Arrays
  - Import NumPy using import numpy as np
  - Create a NumPy array using np.array() or np.arange()
  - NumPy arrays can be one-dimensional or multi-dimensional
- Basic NumPy Functions
  - shape: returns the number of elements in the array
  - zeros: creates an array of zeros
  - ones: creates an array of ones
  - full: creates an array with a specified value
  - array: converts a Python list to a NumPy array
- Accessing Elements in NumPy Arrays is similar to accessing elements in a Python list e.g. arr[0]

SLICING

- Slicing One-Dimensional Arrays:
  - Use array [start : stop] to slice an array
  - start is inclusive, stop is exclusive
  - Use array[start:] to slice from start to the end of the array
  - Use array[:stop] to slice from the beginning of the array to stop
  - Negative indices can be used to count from the end of the array
  - Steps can be used to slice every nth item, e.g. array[::2]
- Slicing Multi-Dimensional Arrays:
  - Use array[row, column] to slice a 2D array
  - row and column can be slices or individual indices
  - Use array[row, :] to slice a entire row
  - Use array[:, column] to slice a entire column

Additional Notes:

- Slicing does not change the original array
- Slicing returns a new array, which is a snapshot of the original array

UNIVERSAL FUNCTIONS

- Examples of UFuncs:
  - np.sqrt:
    - finds the square root of each element
  - np.abs:
    - finds the absolute value of each element
  - np.exp:
    - finds the exponential of each element
  - np.max and np.min:
    - find the maximum and minimum values in an array
  - np.sign:
    - returns the sign of each element (positive, negative, or zero)
  - Trigonometric functions (e.g. np.sin, np.cos, np.tan)
  - Logarithmic functions (e.g. np.log)

COPY VS VIEW

- A "copy" of an array creates a separate and independent array.
- A "view" of an array creates a new array that is connected to the original array.
- Changes made to the original array will be reflected in the view array.
- To create a copy of an array, use the .copy() function.
- To create a view of an array, use the .view() function.
- A copy is completely separate from the original, while a view is always connected both ways.

SHAPE & RESHAPE

- The shape of a numpy array can be found using the shape function, which returns the number of dimensions and the size of each dimension.
- The reshape function is used to change the shape of a numpy array.
- The -1 parameter can be used in the reshape function to automatically determine the size of a dimension.

ITERATION

- iterate through arrays is important for data manipulation.
- ‘ np.nditer() ’ is an easier way to iterate through arrays, especially for higher-dimensional arrays instead of nested loops.

ARRAY SORTING

- The np.sort() function is used to sort numpy arrays in numerical order.
- Numpy arrays can also be sorted alphabetically using the same function.
- When sorting numpy arrays, a copy is returned and the original array is not changed.
- Boolean values can be sorted in numpy arrays, with False coming before True.
- Sorting can also be applied to multi-dimensional numpy arrays, sorting each row individually.

SEARCHING

- The numpy where() function :
  - used to search for items in a numpy array.
  - returns a tuple, and the first element of the tuple is the index number of the element in the array that meet the specified condition not the elements themselves.
  - can return multiple index numbers if there are multiple elements that meet the condition.
  - can also return the elements themselves instead of the index numbers by passing the index number tuple as an argument to the numpy array.
- The numpy zeroth attribute is used to extract the index number from the tuple.
- you can use conditional statements with the numpy where() function to finding even or odd numbers using the modulus operator.
- An example is provided where even numbers are found using the condition numpy array % 2 == 0.

FILTERING

- if I want to filter data from an array you can store filtered data
  - filtered = condition on array you want to filter values Accordingly
  - e.g. filtered = array % 2==0
  - In this example it will filter according to even numbers.

DIFFERENCE BETWEEN RANGE & ARANGE

- ‘range’ is a built-in Python function that returns a range object.
- ‘range’ generates elements lazily.

- ‘arange’ is a NumPy function that returns a NumPy array and allows for specifying non-integer step sizes.
- ‘arange’ generates the entire sequence immediately