**Day 15 Assignment - 20/12/2023 - Vamsi Viswanadham**

Worked on the below Concepts.

NumPy, short for Numerical Python, is an open-source Python library that is widely used for scientific computing. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.

Here are the few numpy functions:

np.array: Creates a new NumPy array.

Usage: np.array([1, 2, 3]) creates an array with elements 1, 2, and 3.

np.arange: Returns evenly spaced values within a given interval.

Usage: np.arange(1, 10, 2) creates an array with values from 1 to 10 (exclusive) with a step of 2 (i.e., [1, 3, 5, 7, 9]).

np.linspace: Returns evenly spaced numbers over a specified interval.

Usage: np.linspace(0, 1, 5) creates an array of 5 numbers evenly spaced between 0 and 1.

np.random.random: Generates random numbers.

Usage: np.random.random((3, 3)) creates a 3x3 array with random numbers between 0 and 1.

np.concatenate: Joins a sequence of arrays along an existing axis.

Usage: np.concatenate((a, b)) concatenates arrays a and b.

np.reshape: Gives a new shape to an array without changing its data.

Usage: np.reshape(d, (9,)) reshapes the array d into a 1D array of 9 elements.

Arithmetic operations (+, -, \*, /): Perform element-wise arithmetic operations on arrays.

Usage: a + 2 adds 2 to each element in the array a.

np.sin: Trigonometric sine, element-wise.

Usage: np.sin(c) computes the sine of each element in array c.

np.dot: Dot product of two arrays.

Usage: np.dot(a, a) computes the dot product of array a with itself.

np.mean: Compute the arithmetic mean along the specified axis.

Usage: np.mean(e) calculates the mean of the elements in array e.

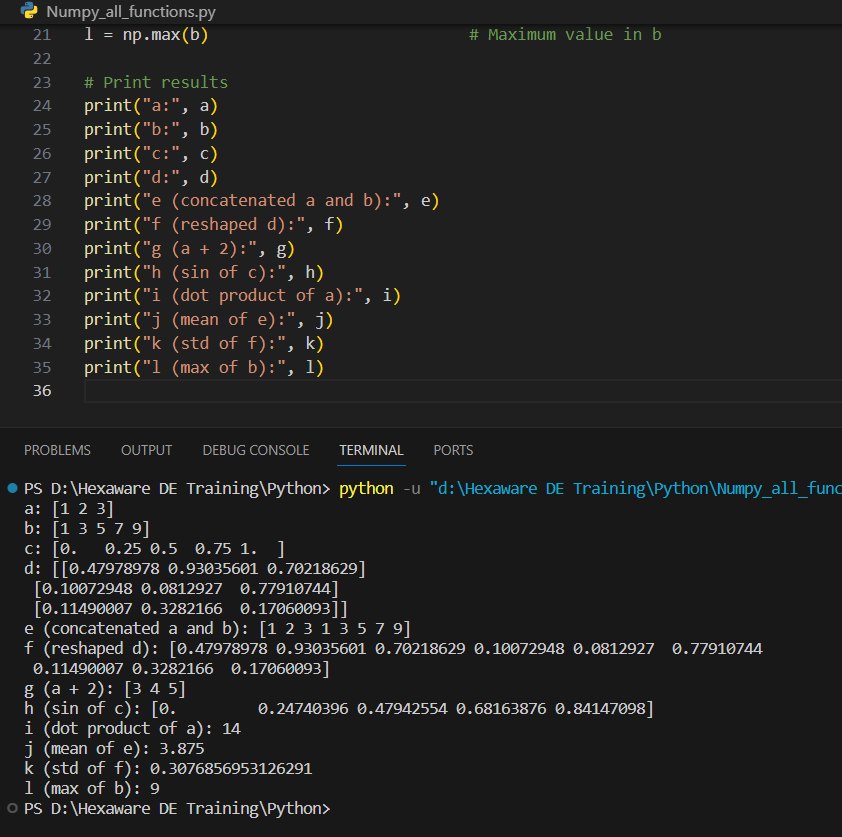
np.std: Compute the standard deviation along the specified axis.

Usage: np.std(f) calculates the standard deviation of the elements in array f.

np.max: Return the maximum along a given axis.

Usage: np.max(b) finds the maximum value in array b.

Here is the code example for the above functions.



Pandas:

Pandas is an open-source Python library widely used for data analysis and manipulation. It's particularly powerful for working with structured data (like tables or time series) and offers a range of functionalities that make it an indispensable tool for data scientists and analysts.

Uses:

DataFrame Creation: Creating a DataFrame from a dictionary.

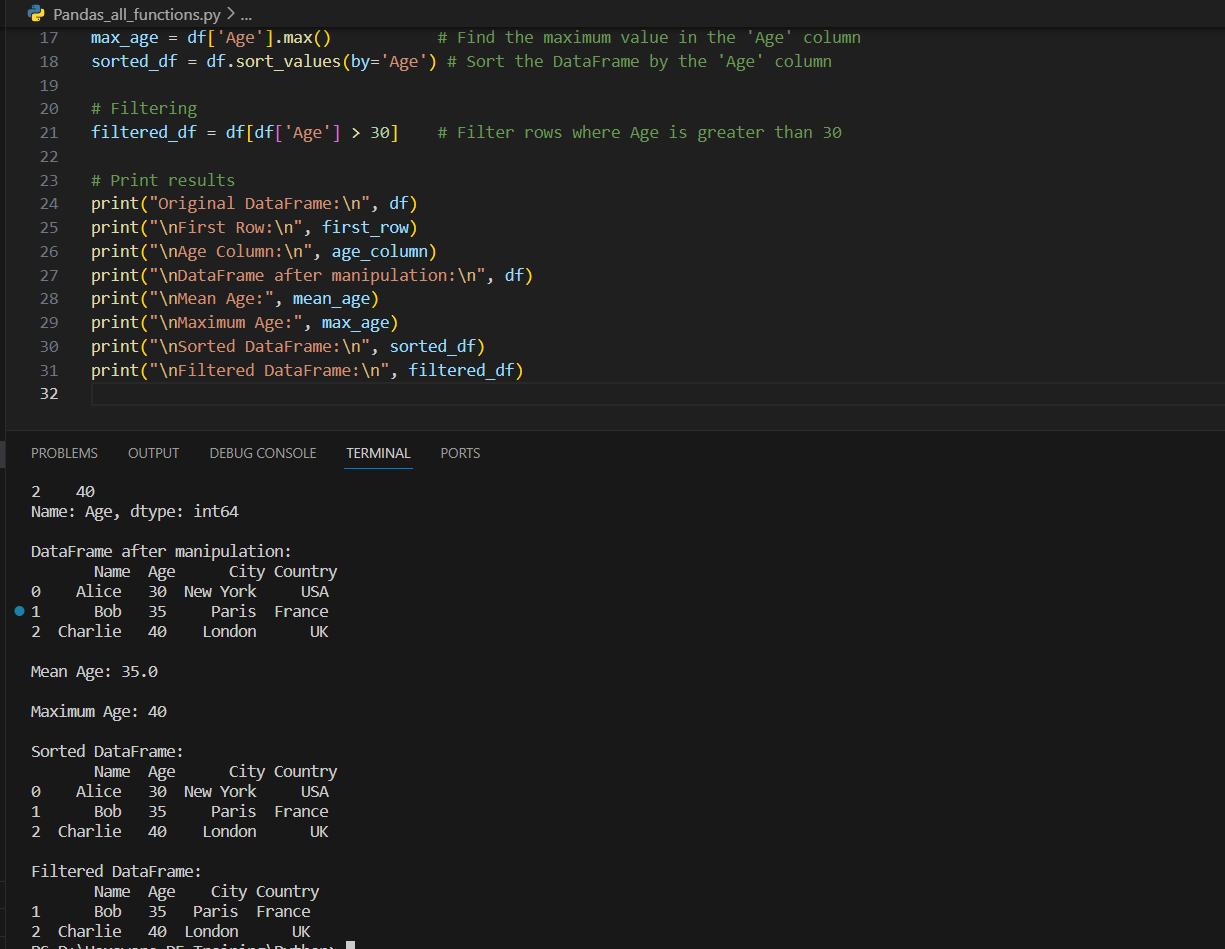
Data Selection: Using .iloc and column selection to access specific rows and columns.

Manipulation: Modifying data by performing operations on columns and adding new columns.

Basic Data Analysis: Calculating statistical measures like mean and maximum, and sorting data.

Filtering: Selecting rows based on a condition.

Here is the output of simple code



Introduction to Big Data and Spark:

Big Data refers to extremely large data sets that are beyond the capability of traditional data processing software to manage and process efficiently. The concept of Big Data is not just about the volume of data but also encompasses the variety of data types and the velocity at which it is generated and processed.

These are often referred to as the Four Vs:

Volume: The sheer amount of data generated from various sources like social media, business transactions, sensors, and so on.

Velocity: The speed at which new data is generated and the pace at which data moves through organizations.

Variety: The different types of data, which can be structured (like databases), semi-structured (like XML files), or unstructured (like text, images, and videos).

Veracity: Reliability of the data feature is considered one of the most important feature of the Big data.

Big Data analytics involves the use of advanced analytic techniques against very large, diverse data sets that include structured, semi-structured, and unstructured data, from different sources, and in different sizes.

Introduction to Apache Spark

Apache Spark is an open-source, distributed computing system that provides a fast and general-purpose cluster-computing framework. Originally developed at the University of California, Berkeley's AMPLab, Spark has seen rapid adoption by enterprises across a wide range of industries.

Its key features include:

Speed: Spark's in-memory data processing capability makes it significantly faster than traditional Hadoop MapReduce for certain applications.

Ease of Use: Spark provides simple-to-use APIs in Python, Java, Scala, and R, making it accessible to a wide audience of developers and data scientists.

Versatility: Spark supports a variety of workloads like batch processing, interactive queries, real-time analytics, machine learning, and graph processing.

Advanced Analytics: Beyond MapReduce, Spark supports SQL queries, streaming data, machine learning, and graph data processing.

Resilient Distributed Datasets (RDDs): Spark introduces the concept of RDDs, which are fault-tolerant collections of elements that can be operated on in parallel. RDDs are the core of Spark’s capabilities, enabling efficient data processing and transformation.