03-03-2025

Training Day – 22

Daily Diary on Data Analysis Topics

WHAT IS DATA ANALYTICS?

sets to extract meaningful insights and support decision-making. This process helps businesses and individuals identify patterns, trends, and actionable conclusions from raw data.

- ➢ **Mathematics and Statistics**: Basics of probability, linear algebra, and hypothesis testing.
- ➢ **Programming**: Learn languages like Python which are widely used for data analysis.
- ➢ **Data Manipulation and Visualization**: Master libraries like:

- *Python: Pandas, NumPy, Matplotlib, Seaborn*
- *R: ggplot2,*

Work on Real-Life Projects

- ○ Start small, such as analyzing public datasets on Kaggle or Google Dataset Search.
- ○ Gradually tackle more complex datasets, like financial records or social media metrics

Tools for Data Analytics

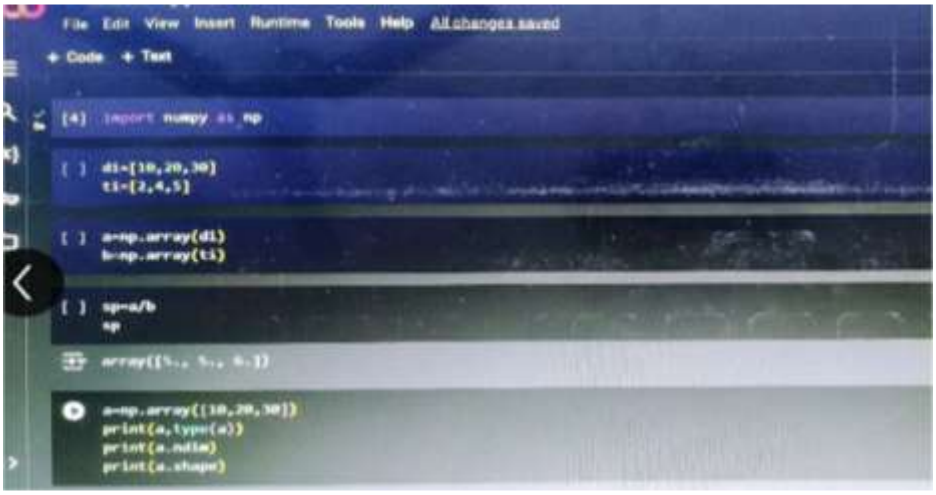
1. *Data Processing and Analysis*
   - ○ **Excel**: Good for small-scale analysis.
   - ○ **Python**: Libraries like Pandas, NumPy, and Scikit-learn.
2. *Data Visualization*
   - ○ **Tableau**: Easy-to-use for creating interactive dashboards.
   - ○ **Power BI**: Microsoft’s tool for creating visual reports.
   - ○ **Matplotlib and Seaborn**: Python-based libraries for visualization.

Topic: Introduction to NumPy Variables

- Learned about numpy.ndarray, its creation, and basic properties.
- Example: Created arrays using np.array() and explored their dimensions, shapes, and data types learned Numpy (np) library and some in-built functions of numpy.

  Functions like -> *astype, size, ndim, dtype, shape, type()*

  •And also practiced about indexing and slicing of arrays in numpy.



```
[4] import numpy as np
```

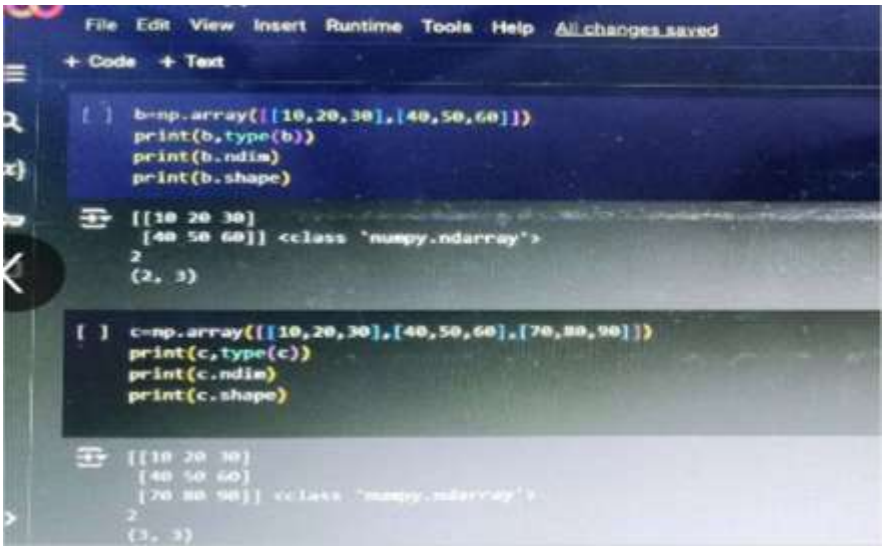
```
[ ] d1=[10,20,30]
    t1=[2,4,5]
```

```
[ ] a=np.array(d1)
    b=np.array(t1)
```

```
[ ] sp=a/b
    sp
```

```
array([5., 5., 6.])
```

```
a=np.array([10,20,30])
print(a,type(a))
print(a.ndim)
print(a.shape)
```



```
[ ] b=np.array([[10,20,30],[40,50,60]])
    print(b,type(b))
    print(b.ndim)
    print(b.shape)
```

```
[[10 20 30]
 [40 50 60]] <class 'numpy.ndarray'>
2
(2, 3)
```

```
[ ] c=np.array([[10,20,30],[40,50,60],[70,80,90]])
    print(c,type(c))
    print(c.ndim)
    print(c.shape)
```

```
[[10 20 30]
 [40 50 60]
 [70 80 90]] <class 'numpy.ndarray'>
2
(3, 3)
```

04-03-2025

Training Day – 23

Topic:- NumPy Manipulation

- Performed reshaping, slicing, and broadcasting operations.
- Example: Reshaped a 1D array into a 2D array and performed element-wise multiplication.

Day-2

- Today I practiced some example based on slicing in multi-dimensional array.
- Some new topics also covered in today's session

as like -

*Shape Manipulation.

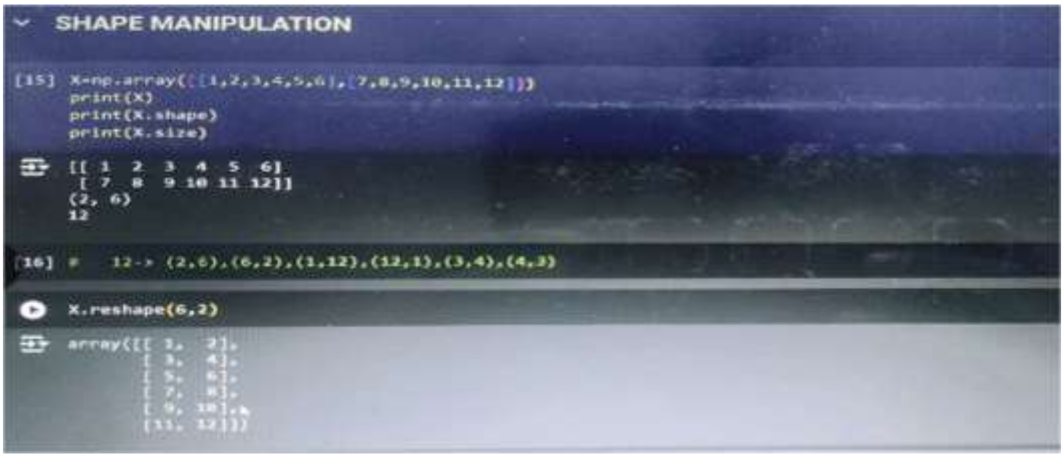
*Conditions In Array.

- And also getting a short knowledge of "Broadcasting" using numpy's library.

NumPy arrays can be reshaped, sliced, and indexed to suit various needs.

EXAMPLE

```
array = np.arange(10)
reshaped =
array.reshape(2, 5) sliced =
reshaped[:, 1:4]
broadcasted = reshaped +
10
print("Original Array:\n", array)
print("Reshaped Array:\n", reshaped)
print("Sliced Array:\n", sliced)
print("Broadcasted Array:\n", broadcasted)
```



SHAPE MANIPULATION

```
[15] X=np.array([[1,2,3,4,5,6],[7,8,9,10,11,12]])
     print(X)
     print(X.shape)
     print(X.size)

     [[ 1  2  3  4  5  6]
      [ 7  8  9 10 11 12]]
     (2, 6)
     12

[16] #   12-> (2,6),(6,2),(1,12),(12,1),(3,4),(4,3)

     X.reshape(6,2)

     array([[ 1,  2],
            [ 3,  4],
            [ 5,  6],
            [ 7,  8],
            [ 9, 10],
            [11, 12]])
```

05-03-2025

Training Day – 24

Topic: Introduction to SciPy

SciPy is an open-source Python library used for scientific and technical computing. It builds on NumPy, providing a wide range of functionalities for mathematics, science, and engineering. SciPy includes modules for optimization, integration, interpolation, eigenvalue problems, signal processing, linear algebra, and more. It is designed to work efficiently with NumPy arrays, allowing users to perform complex computations with minimal code.

- Explored optimization and integration functions in SciPy.
- Example: Used scipy .integrate. quad for numerical integration.
- SciPy builds on NumPy and provides additional modules for optimization, integration, and statistics.

Key Features of SciPy

1. **Linear Algebra**: Tools for solving linear systems, eigenvalues, and singular value decompositions.
2. **Optimization**: Algorithms for optimization, including curve fitting and minimization.
3. **Integration**: Functions for numerical integration.
4. **Interpolation**: Methods for data interpolation.
5. **Statistics**: A wide range of statistical functions and random distributions.
6. **Signal Processing**: Tools for filtering, spectral analysis, and more.

EXAMPLE

```
from scipy.integrate import quad
def func(x):
    return x**2
result, error = quad(func, 0, 1)
print("Integral of x^2 from 0 to 1:", result)
```

06-03-2025

Training Day – 25

Topic: Introduction to Pandas

- Learned about Pandas DataFrame and Series.
- Example: Created a DataFrame from a dictionary and accessed its rows and columns. Pandas provides Series and DataFrame structures for efficient data manipulation and analysis.

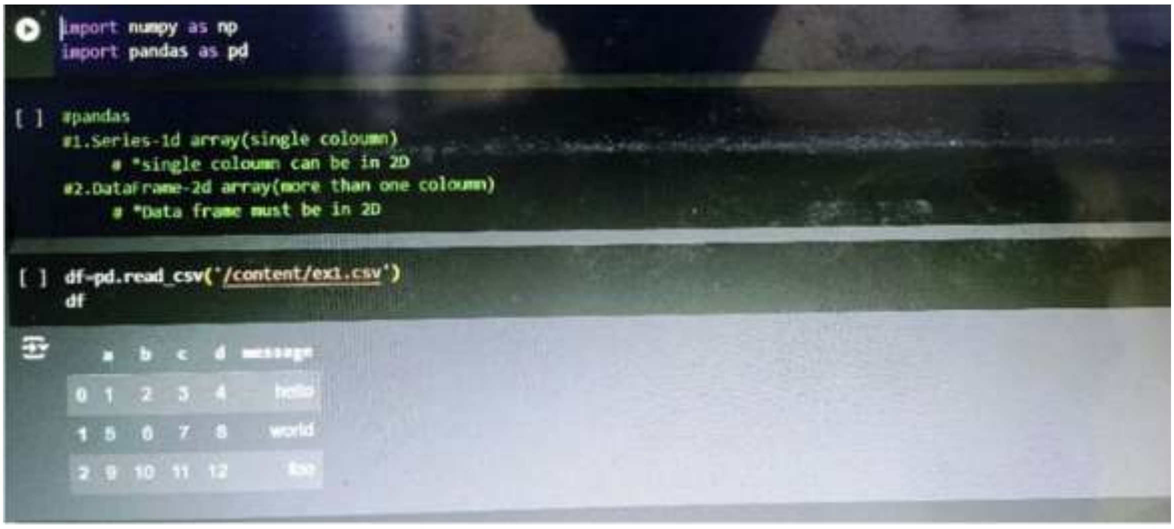
In today's session we worked little bit more on Broadcasting on numpy library,by solving a mathematical operation on matrices.

- Also learn about "fromfunction" through the help of numpy.
- Topic Array creation also covered in today's session, and completed with solving one example of 'Array Creation' using static method.

EXAMPLE

```
import pandas as pd
data = {"Name": ["Alice", "Bob"], "Age": [25, 30]}
df = pd.DataFrame(data)
print(df)
```

- **After importing I used some functions of pandas to read and analys the data set, like - . shape,. isnull(),.info,. isnull(). sum(),.head(),. tail()**

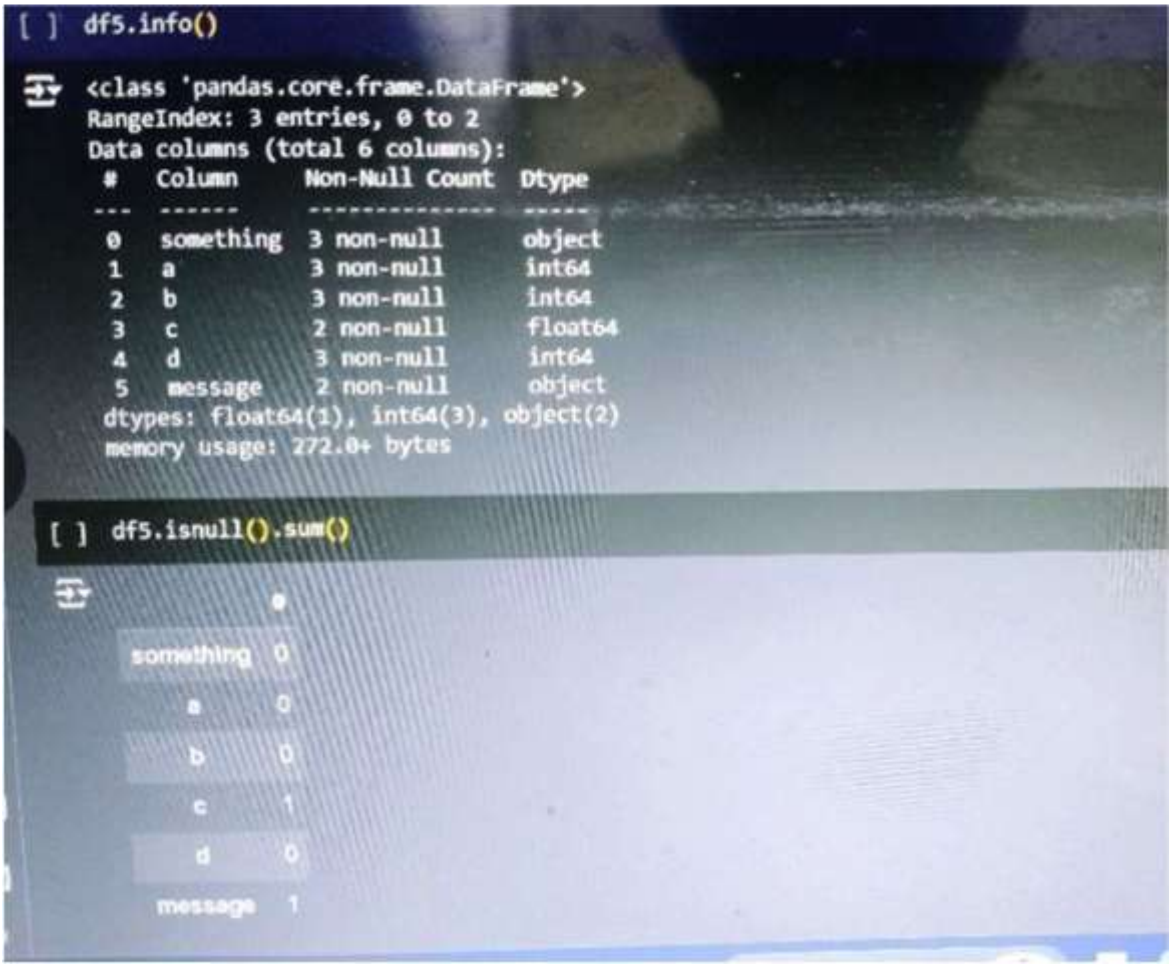


```
import numpy as np
import pandas as pd
```

```
#pandas
#1.Series-1d array(single coloumn)
     # *single coloumn can be in 2D
#2.DataFrame-2d array(more than one coloumn)
     # *Data frame must be in 2D
```

```
df-pd.read_csv('/content/ex1.csv')
df
```

```
   a  b   c   d  message
0  1  2   3   4    hello
1  5  6   7   8    world
2  9  10  11  12     foo
```



```
[ ] df5.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3 entries, 0 to 2
Data columns (total 6 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   something  3 non-null      object
 1   a          3 non-null      int64
 2   b          3 non-null      int64
 3   c          2 non-null      float64
 4   d          3 non-null      int64
 5   message    2 non-null      object
dtypes: float64(1), int64(3), object(2)
memory usage: 272.0+ bytes
```

```
[ ] df5.isnull().sum()
```

```
           0
something  0
        a  0
        b  0
        c  1
        d  0
  message  1
```

07-03-2025

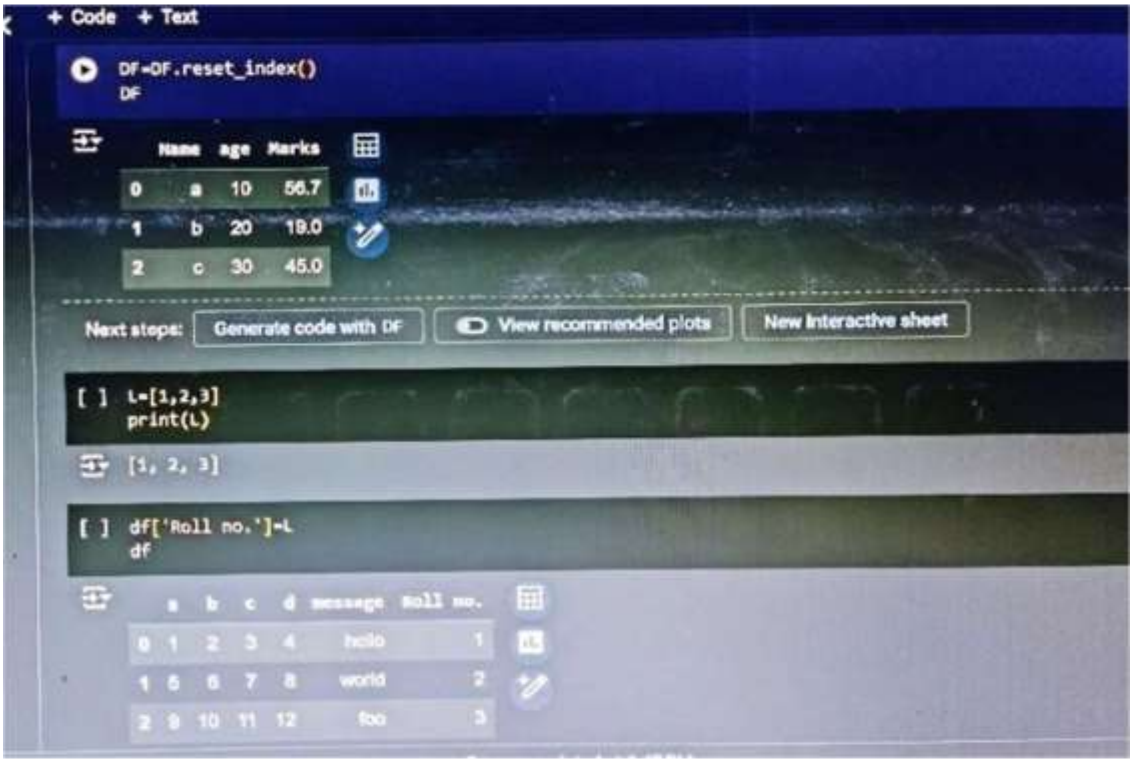
Training Day – 26

- **Topic: Descriptive Analysis with Pandas**
  - Summarized data using .describe(), .mean(), and .sum().
  - Example: Analyzed a dataset's central tendencies and spread.

Today I made my own data set and convert it into DataFrame and apply some functions of **pandas library.** Like - .info(), .set_index(), .reset_index(), adding an column name 'Roll no.' ,.drop().

• Also done 'Indexing' and 'Slicing' on large dataset for reading the data according to given rows and column detail, using '.iloc' function.

• Also practice for feeling NAN values by identifying outliers, with the help of mean, median,mode.



```
DF=DF.reset_index()
DF
```

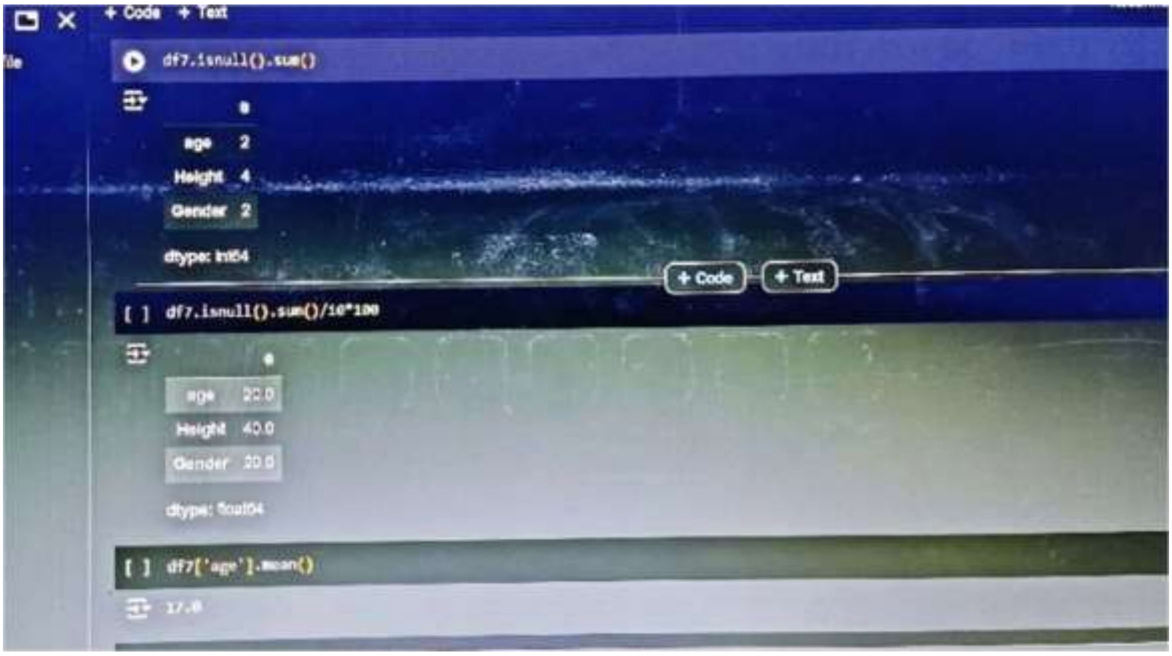
	Name	age	Marks
0	a	10	56.7
1	b	20	19.0
2	c	30	45.0

```
[ ] L=[1,2,3]
    print(L)
```

```
[1, 2, 3]
```

```
[ ] df['Roll no.']=L
    df
```

	a	b	c	d	message	Roll no.
0	1	2	3	4	hello	1
1	5	6	7	8	world	2
2	9	10	11	12	foo	3



```
df7.isnull().sum()
```

```
        0
age     2
Height  4
Gender  2

dtype: int64
```

```
[ ] df7.isnull().sum()/10*100
```

```
        0
age     20.0
Height  40.0
Gender  20.0

dtype: float64
```

```
[ ] df7['age'].mean()
```

```
17.0
```

df = pd.DataFrame({"Values": [10, 20, 30, 40, 50]}) print(df.describe())