[Day-13]

Data Analysis with Python

1) Numpy: Scientific computing in Python

↳ Provides support for arrays & matrices alongwith a collection of mathematical functions to operate on these data structures.

```
import numpy as np
arr1 = np.array([1,2,3,4])
print(arr1)

np.arange(0,10,2)   # array([0,2,4,6,8]) : o/p

np.arange(0,10,2).reshape(5,1)   # array([[0],   : o/p
                                         [2]
                                         [4]
                                         [6]
                                         [8]])

np.eye(3)  # identity matrix
```

Addn : arr1 + arr2

Subtractn : arr1 - arr2 ... so on ...

```
(np.sqrt(arr))   # squareroot
(np.exp(arr))    # exponent
```

→ used for slicing & indexing as well
→ modifying array elements ~
→ Statistical concepts — np.mean(), np.std() - - etc. ~
→ logical operations ~

2) Pandas: data manipulation library in Python
→ used for data analysis & data cleaning
→ Provides 2 primary data structures: series & dataframe
(one dimensional array like object)
└ 2 dimensional, size-mutable & potentially heterogeneous tabular data structure with labeled axes.

→ import pandas as pd

```
data = [1,2,3]
series = pd.series(data)
```

```
data = { 'Name': ['Ram', 'Tom', 'Kim'],
         'Age': [12, 14, 18],
         'City': ['Agra', 'Pune', 'Florida']
       }
df = pd.DataFrame(data)
# Accessing a specified element : df.at[2,'Name']
# Rename column : df.drop('City')
```

→ Pandas provides a wide range of functions for data manipulation & analysis, making it easier to clean, transform, and extract insights from data.

Data Visualization with Matplotlib

→ Plotting library that enables the creation of static, animated, and interactive visualizations

→ widely used in data visualizatn in data science & analytics.

```
import matplotlib.pyplot as plt
```

line plot : plt.plot(x,y)

Basic :
```
plt.xlabel('X axis')
plt.ylabel('Y axis')
plt.title("Basic Line Plot")
plt.show()
```

Format : plt.plot(x, y, color='red', linestyle='--', marker='o', linewidth=3, markersize=8)

plt.grid(True) : for grid

Multiple Plots :
```
plt.subplot(2,2,1)
plt.plot(x, y1, color='red')
plt.title("Plot 1")

plt.subplot(2,2,2)
plt.plot(y1, x, color='blue')
plt.title("Plot 2")
```

Histogram : plt.hist(data, bins=5) color='orange' edgecolor='red'

Scatter Plot : plt.scatter(x, y, color='red' marker='o')

Data visualization with Seaborn

→ Python visualizatn library based on Matplotlib that provides a high-level interface for drawing attractive and informative statistical graphics.

→ Seaborn helps in creating complex visualizations

```
import seaborn as sns
```

Scatter plot : sns.scatterplot(x, y)

Line plot : sns.lineplot(x, y)

Barplot : sns.barplot(x='', y='', data=data1)

Box plot : sns.boxplot( )

Violinplot : sns.violinplot( )

Histogram : sns.histplot(data1, bins=10, kde=True)

kdeplot : sns.kdeplot( )

Pairplot : sns.pairplot( )

Heatmap : sns.heatmap(corr, annot=True, cmap='coolwarm')