

PART 1: MONTH 1 – PYTHON BASICS AND DATA MANIPULATION

Week 3: NumPy and Pandas for Data Manipulation

◆ Theory (Study Part)

1. NumPy

- **Arrays** → like lists but faster and better for numerical computation.

```
• import numpy as np
• import pandas as pd
• np.array([1,2,3,4,5])
```

- **Operations** → element-wise calculations.

```
print("Multiply by 2 ",data * 2)
print("Multiply by 5 ", data* 5)
```

- **Broadcasting** - applying operations between arrays of different shapes.

```
• import numpy as np
• import pandas as pd
• #Broadcasting
• a = np.array([1,2,3])
• b = 2
• print(a + b)    # [3 4 5]
```

2. Pandas

- **Series** → one-dimensional labeled array.

```
• import pandas as pd
• s = pd.Series([10, 20, 30], index=["a","b","c"])
•
```

- **DataFrame** → 2D table of rows and columns (like Excel).

```
• import pandas as pd
• data = {"Name":["Mansi","Raj"], "Age":[21,22]}
• df = pd.DataFrame(data)
•
```

- **Indexing** → selecting data.

```
• import pandas as pd
• data = {"Name":["Mansi","Raj"], "Age":[21,22]}
```

- `print(df["Name"])` # column
- `print(df.loc[0])` # row by label
- `print(df.iloc[1])` # row by index
-

- **Grouping** → aggregate/group data.

- `df.groupby("Age").mean()`

Hands – On

1.Numpy -Example

```
import numpy as np
import pandas as pd
data=np.array([1,2,3,4,5])
mean=np.mean(data)
print(mean)
print("Max : ", np.median(data))
print("Total",np.sum(data))
print("Square : ",np.sqrt(data))
print("Multiply by 2 ",data * 2)
print("Multiply by 5 ", data* 5)
```

2.Pandas – Example

```
data={"Name" : ["Mansi","Gaurav","Seema","Radha","Kishna"],
      "Age" : [19, 19, 22, 22,23],
      "Marks" : [89,89,76,90,54],
      "City" :["Indore","Mumbai","Jaipur","Indore","Mumbai"]}

df=pd.DataFrame(data)
print("Original Dataset : ")
print(df)
print("-----")
print("Name Column:")
print(df['Name'])
print("-----")
print("First Row ")
```

```
print(df.loc[0])
print("-----")
print("Average Marks by Age-")
print(df.groupby('City')['Marks'].mean())
```

Client Project – Data Cleaning + Aggregation

```
import pandas as pd
import numpy as np
#Client Project = Data Cleaning + Aggregation
data = {"Student":
["Mansi", "Gaurav", "Mansi", "Gaurav", "Reema"],
        "Subject": ["Math", "Math", "Sci", "Sci", "Math"],
        "Marks": [90, 80, None, 90, 80]}

df=pd.DataFrame(data)
print("Original Data")
print(df)
cln_dt=df.dropna()
print(cln_dt)
print("Unduplicated---")
print(cln_dt.drop_duplicates())
print("_____")
print("Grouping ")
print(cln_dt.groupby('Student')['Marks'].mean())
```

Summary (for Submission)

Summary – Week 3: NumPy and Pandas for Data Manipulation

In Week 3, I studied NumPy and Pandas, two powerful Python libraries for data analysis.

NumPy

- Provides arrays for fast numerical operations.
- Supports element-wise operations (addition, multiplication, etc.).
- Broadcasting allows operations between arrays of different shapes.
- Functions like np.mean(), np.max(), etc. simplify calculations.

Pandas

- Series → one-dimensional labeled data.
 - DataFrame → two-dimensional data table (like Excel).
 - Indexing with `.loc[]` and `.iloc[]` for selecting rows/columns.
 - Grouping with `groupby()` for aggregation.
-

◆ Hands-On

- Performed array operations with NumPy.
 - Created and manipulated Pandas DataFrames.
 - Used indexing, filtering, and grouping.
-

◆ Client Project

I developed a data cleaning and aggregation script that:

1. Removed missing values.
2. Removed duplicates.
3. Calculated average marks per student using `groupby()`.