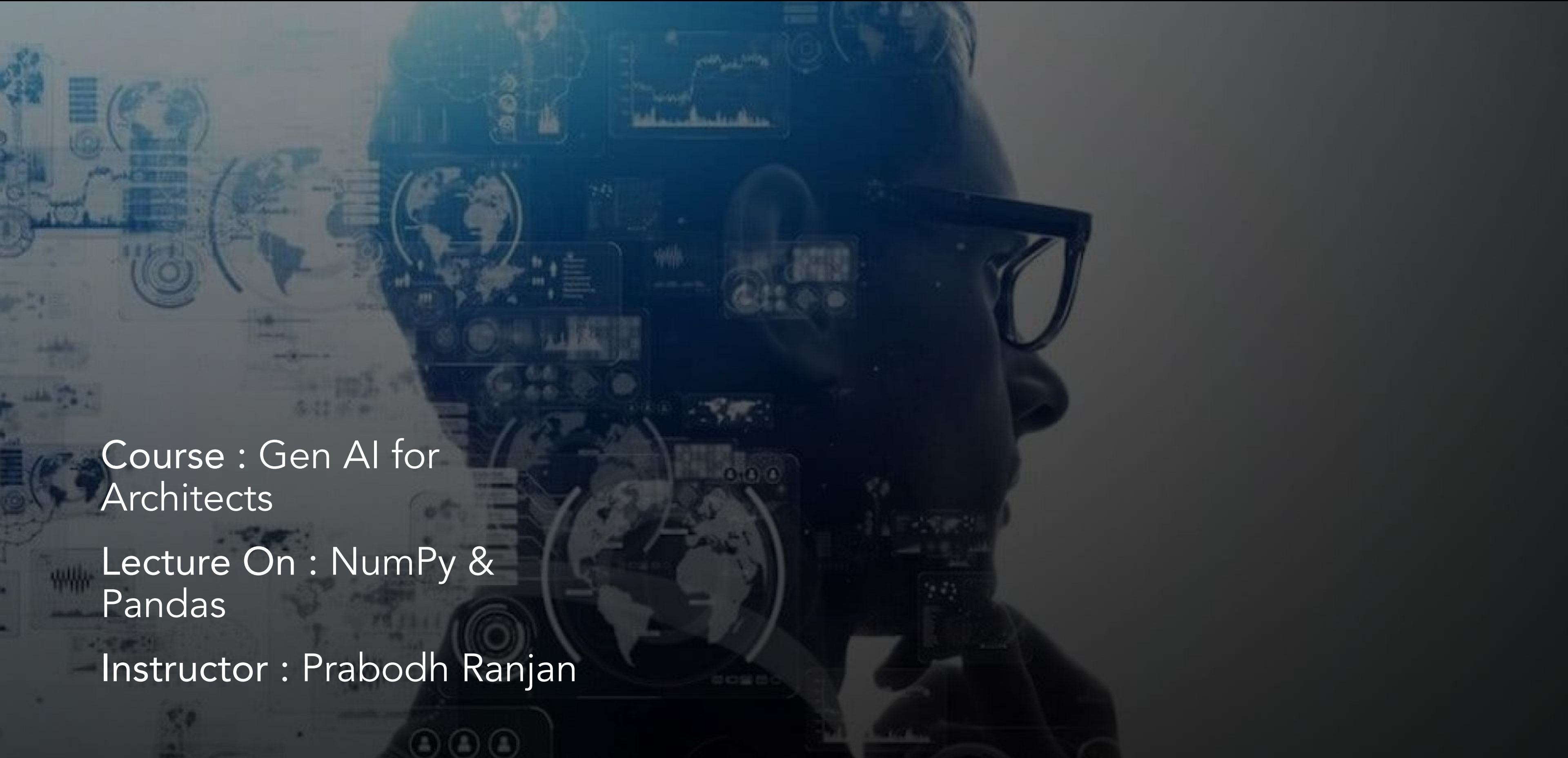
upGrad ENTERPRISE

GENERATIVE AI FOR ARCHITECTS

July 2024

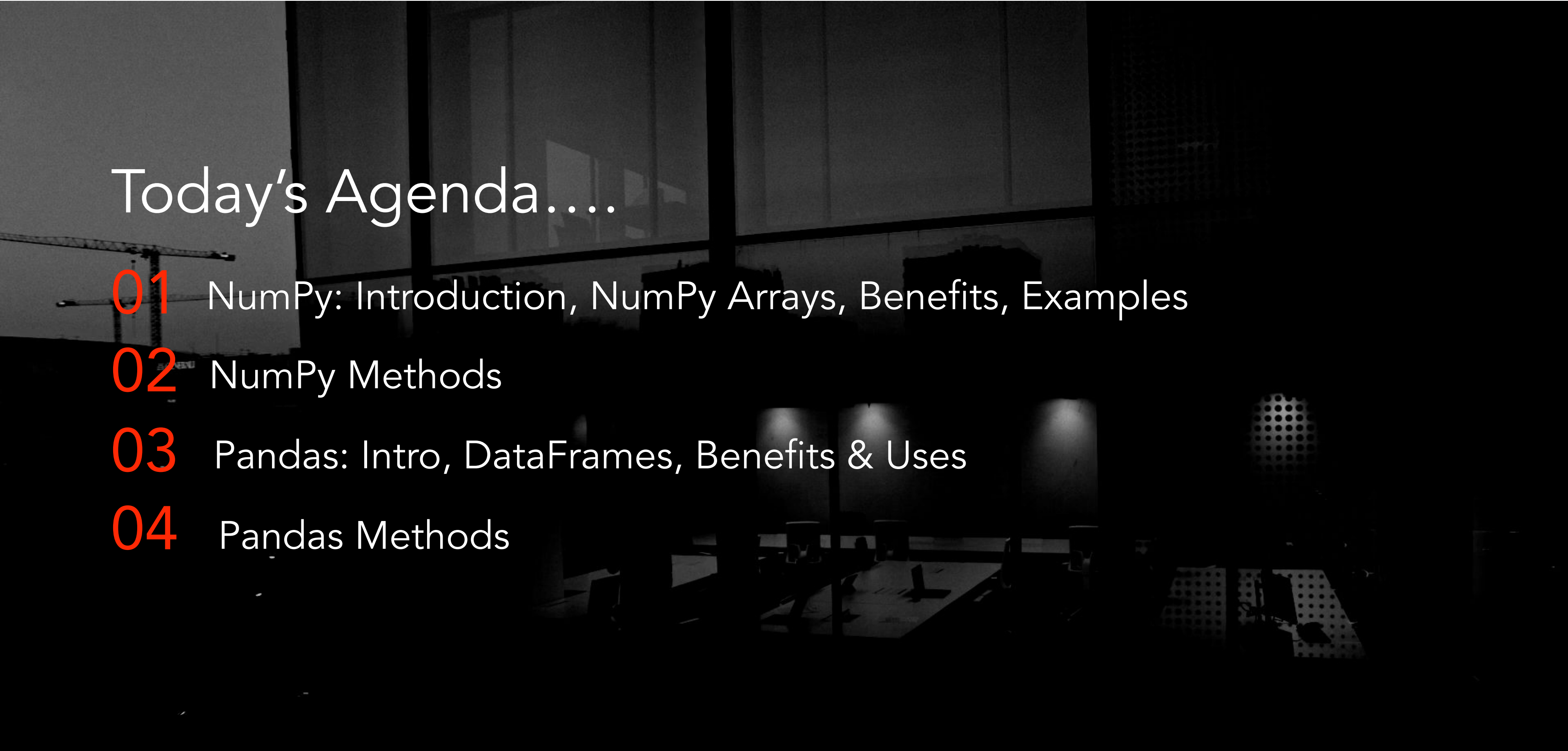
upGrad ENTERPRISE



Course : Gen AI for Architects

Lecture On : NumPy & Pandas

Instructor : Prabodh Ranjan



Today's Agenda….

01 NumPy: Introduction, NumPy Arrays, Benefits, Examples

02 NumPy Methods

03 Pandas: Intro, DataFrames, Benefits & Uses

04 Pandas Methods

upGrad ENTERPRISE

NumPy

What is NumPy?

- NumPy is a Python library used for analyzing data.
- Essential for data science, providing fast operations on arrays due to its bindings to C libraries.
- Used in various mathematical, statistical, and machine learning applications.

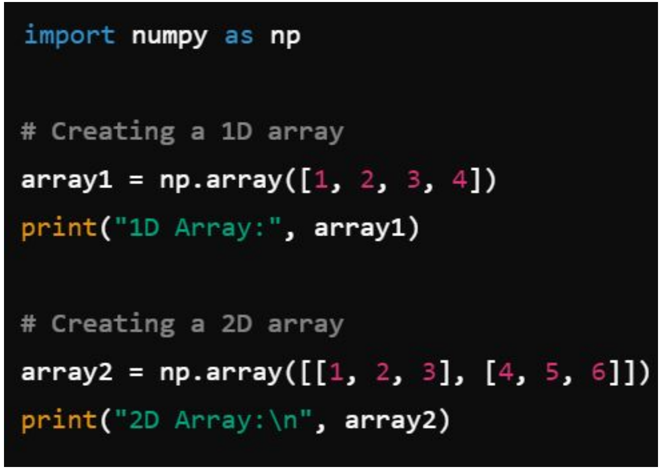
REFERENCE: NumPy Documentation

NumPy Arrays:

- NumPy Arrays are the fundamental data structure of the NumPy library.
- They are grid-like data structures that hold elements of the same type, allowing for efficient mathematical operations.

Creating a NumPy Array:

Code Input:

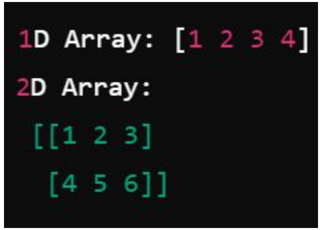


```python
import numpy as np

# Creating a 1D array
array1 = np.array([1, 2, 3, 4])
print("1D Array:", array1)

# Creating a 2D array
array2 = np.array([[1, 2, 3], [4, 5, 6]])
print("2D Array:\n", array2)
```

Output:



```
1D Array: [1 2 3 4]
2D Array:
 [[1 2 3]
  [4 5 6]]
```

Importance of NumPy Arrays in Machine Learning:

- Efficiency: NumPy arrays are highly efficient for mathematical and statistical operations, thanks to their underlying implementation in C.

- Ease of Use: Provides a wide range of functions for array creation, manipulation, and mathematical computations.

- Integration: Seamlessly integrates with other libraries like pandas, SciPy, and matplotlib, making it a cornerstone of the scientific Python ecosystem.

- Versatility: Supports various types of operations, including element-wise operations, broadcasting, and more, which are essential for data analysis and machine learning tasks.

NumPy Arrays - Matrices

Matrices

- Matrices are a type of NumPy array that are two-dimensional.
- They consist of rows and columns.
- Even if they have only one row or one column, they are still considered matrices.
- Commonly used in various mathematical operations such as matrix multiplication, linear transformations, and more.

NumPy Arrays - Vectors

Vectors

- Vectors are another type of NumPy array that are strictly one-dimensional.
- They can be thought of as lists of numbers, representing a point in space.
- Vectors are often used in operations such as dot products, cross products, and in solving systems of linear equations.

NumPy
Matrices vs Vectors

Matrices

- Linear Algebra: Used extensively in linear algebra for operations such as matrix multiplication, inversion, and finding eigenvalues.

- Data Representation: In machine learning, matrices are often used to represent datasets, where rows correspond to samples and columns correspond to features.

- Image Processing: Images can be represented as matrices of pixel values.

- Graph Theory: Adjacency matrices represent the connections between nodes in a graph.

Vectors

- Spatial Representation: Vectors are used to represent points or directions in space. For example, a vector (x, y, z) in 3D space.

- Feature Vectors: In machine learning, feature vectors represent the attributes of data points.

- Physics and Engineering: Vectors represent quantities that have both magnitude and direction, such as velocity and force.

- Linear Algebra Operations: Vectors are used in operations like dot product and cross product.

LET'S CONTINUE IN GOOGLE COLAB FOR HANDS-ON CODING!

Pandas

What is Pandas?

- **Pandas** is an open-source data manipulation and analysis library for Python.

- Designed to handle, clean, and analyze data efficiently.

- Allows loading data from various sources, transforming it, and performing complex operations with ease.

REFERENCE: Pandas Documentation

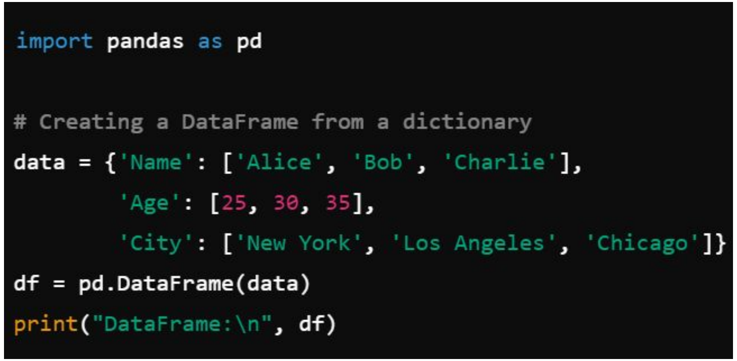
Creating a DataFrame:

- Pandas primarily works with two data structures: Series and DataFrames

- A DataFrame is like a spreadsheet, with rows and columns.

Creating a DataFrame:

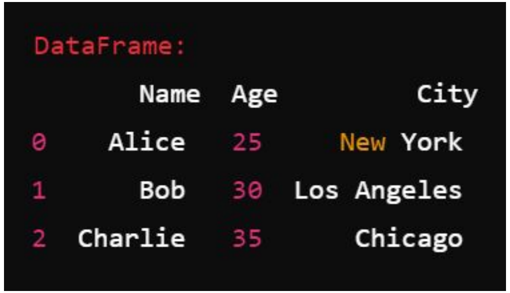
Code Input:



```
import pandas as pd

# Creating a DataFrame from a dictionary
data = {'Name': ['Alice', 'Bob', 'Charlie'],
        'Age': [25, 30, 35],
        'City': ['New York', 'Los Angeles', 'Chicago']}
df = pd.DataFrame(data)
print("DataFrame:\n", df)
```

Output:



```
DataFrame:
       Name  Age         City
0    Alice   25     New York
1      Bob   30  Los Angeles
2  Charlie   35      Chicago
```

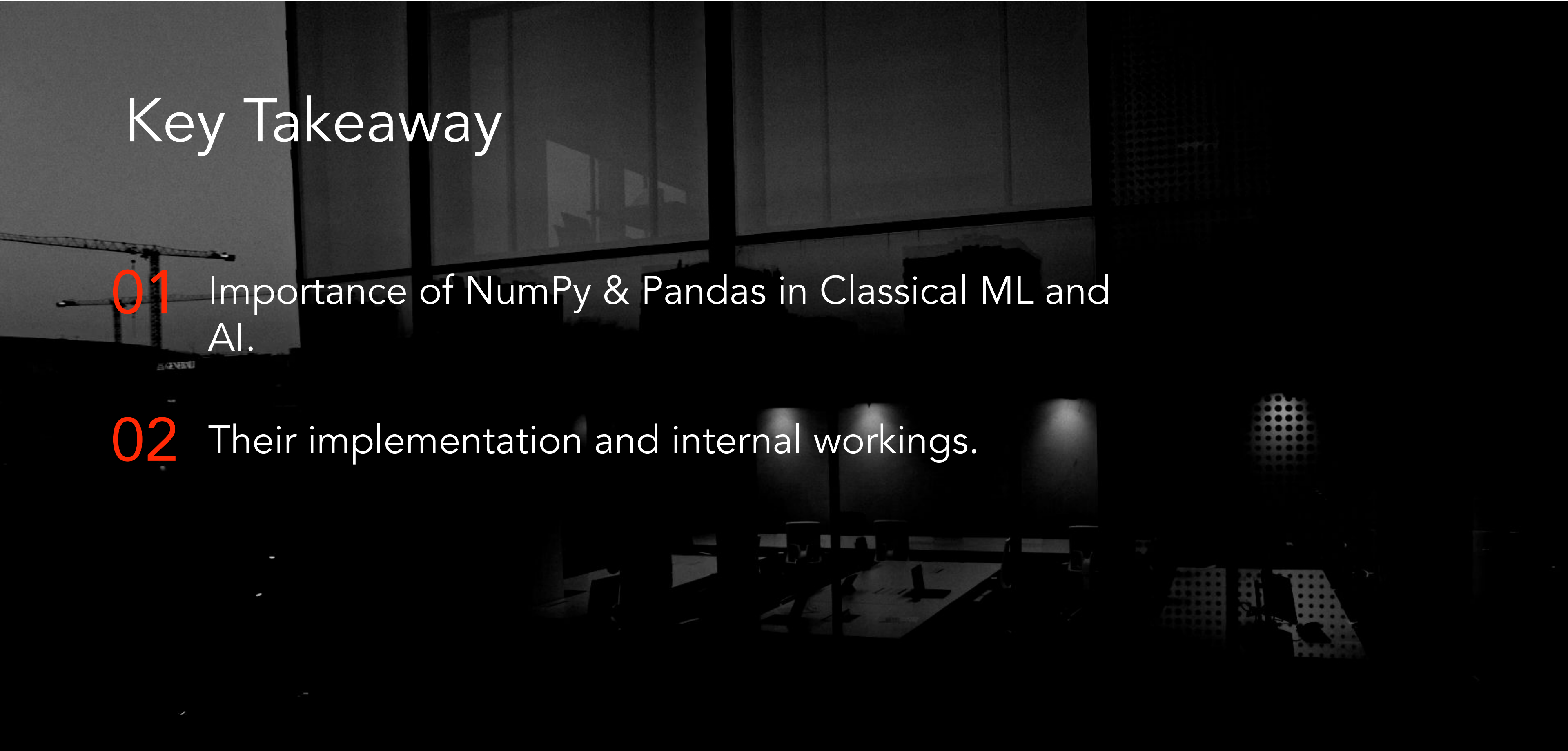
Importance of NumPy Arrays in Machine Learning:

- Structured Data Handling: Tabular structure similar to a spreadsheet.
- Flexibility: Handles various data types (numerical, textual, datetime).
- Easy Data Import/Export: Reads from and writes to CSV, Excel, SQL databases, etc.
- Data Cleaning: Handles missing data, removes duplicates, transforms data.
- Data Selection: Selects specific rows and columns based on conditions.
- Data Aggregation: Groups and aggregates data efficiently.
- Data Visualization: Integrates with Matplotlib and Seaborn for plotting.
- Time Series Data: Robust tools for working with date and time data.
- Data Joining/Merging: Combines multiple DataFrames.
- Efficient Memory Usage: Optimized for large datasets.
- Machine Learning Prep: Prepares data for machine learning tasks.
- Interactive Analysis: Allows for step-by-step data exploration in Jupyter Notebooks.
- Community and Documentation: Extensive support and resources available.

Use Cases of Pandas:

- **Data Cleaning:** Clean messy data, remove duplicates, handle missing values, and transform data into a structured format.
- **Data Analysis:** Perform in-depth data analysis, calculate statistics, visualize data, and make data-driven decisions.
- **Data Preparation:** Prepare data for machine learning by encoding categorical variables and splitting data into training and testing sets.

LET'S CONTINUE IN GOOGLE COLAB FOR HANDS-ON CODING!



Key Takeaway

01 Importance of NumPy & Pandas in Classical ML and AI.

02 Their implementation and internal workings.

upGrad ENTERPRISE