**NumPy for Data Analytics**

**Duration**: 4 Hours  
**Prerequisites**: Basic Python programming

**Objective:**

* Confidently use NumPy for structured numeric data analysis
* Build and manipulate arrays to extract trends and patterns
* Perform comparisons, filtering, and transformations in reusable, scalable ways
* Be well-prepared to work with real-world datasets (e.g., COVID-19 time-series, economic indicators, sales trends)

**Hour 1: NumPy Foundations for Structured Data**

**Goal**: Understand core concepts of NumPy arrays and basic data manipulation

**Topics**:

* Introduction to NumPy for analytical workflows
  + Why use NumPy for large-scale, structured data
  + Differences between Python lists and NumPy arrays
* Array creation & initialization
  + Creating 1D and 2D arrays from lists or sequences
  + Functions: np.array(), np.arange(), np.linspace(), np.zeros(), np.ones(), np.full()
* Exploring array structure
  + Attributes: shape, ndim, dtype, size
  + Conceptualizing rows as categories/entities and columns as time-based observations (or vice versa)
* Indexing and slicing
  + Row and column slicing
  + Subsetting blocks of rows/columns
  + Reversing and skipping elements with slicing

**Exercise**:

* Create and slice arrays representing simple time-series or categorical data

**Hour 2: Numerical Operations & Conditional Logic**

**Goal**: Perform arithmetic, aggregations, and apply conditional logic to arrays

**Topics**:

* Arithmetic and element-wise operations
  + Additive and multiplicative operations across arrays
  + Per-unit comparisons and change detection
* Aggregations & summaries
  + Using sum, mean, median, std, max, min
  + Applying aggregations along different axes
* Boolean arrays and conditional selection
  + Creating condition masks
  + Filtering with single or multiple conditions
  + Using np.where() for conditional assignments or queries
* Handling missing values
  + np.nan, np.isnan(), np.nanmean(), np.nan\_to\_num()

**Exercise**:

* Compute summary statistics and filter datasets based on conditions
* Handle missing data in arrays

**Hour 3: Reshaping, Combining, and Sorting Data**

**Goal**: Reshape and reorganize data for flexible analysis

**Topics**:

* Reshaping and flattening
  + Using reshape(), ravel(), and flatten() for structural adjustments
  + Preparing time-series or cross-sectional data for processing
* Concatenating and splitting arrays
  + Combining with np.concatenate(), np.vstack(), np.hstack()
  + Splitting with np.split(), np.hsplit(), np.vsplit()
* Advanced indexing
  + Integer array indexing and slices for targeted selection
  + Fancy indexing to retrieve or reorder specific records
* Sorting and ranking
  + Sorting by values with np.sort()
  + Finding ranks or index orders with np.argsort()

**Exercise**:

* Reshape and concatenate multiple arrays
* Sort data and perform fancy indexing

**Hour 4: Deriving Insights & Data Processing Workflows**

**Goal**: Compute derived metrics, analyze relationships, and manage data pipelines

**Topics**:

* Derived metrics and computation
  + Row- and column-wise calculations
  + Calculating relative changes: (x[i] - x[i-1]) / x[i-1]
  + Computing percentage rates and ratios
* Correlation and linear operations
  + Using np.corrcoef() to evaluate variable relationships
  + Matrix operations with dot() for aggregated metrics
* Sampling and randomization
  + Generating random samples and synthetic data
  + Setting seeds for reproducibility
* Saving and loading arrays
  + Using np.save(), np.load(), np.savetxt() for data persistence and reuse

**Mini Project**:

* Analyze a real-world dataset (e.g., COVID-19 time-series or sales data) using NumPy
* Compute key metrics, correlations, and transformations
* Save processed arrays for further analysis