

# NumPy Roadmap 2026 – 2 Months (Week 1 to Week 8)

*A complete, practical, beginner-to-intermediate learning path for Data Science, ML & Scientific Computing*

- **Designed by:** Senior Data Scientist & Python Educator (5+ years experience)
- **Outcome after 8 weeks:**  
You'll confidently use NumPy for real-world data analysis, performance-efficient computations, ML preprocessing, and interviews.

---

## Roadmap Overview

Duration	Level	Weekly Time	Outcome
8 Weeks	Beginner → Intermediate	6-8 hrs/Week	Job-ready NumPy skills

## Prerequisites

- Python basics (loops, functions, lists)
  - High-school level math (mean, matrix idea)
  - No prior NumPy experience required
-

# Week 1 – NumPy Foundations & ndarray Core

## Weekly Goal

Understand why NumPy exists, how ndarray works, and how it differs from Python lists.

---

## Key Concepts & Theory

- Why NumPy over Python lists
- Installing & importing NumPy
- ndarray structure
- Shape, dimensions, and dtype
- Vectorized operations (conceptual)

## Important Functions & Methods

Concept	Functions
Create array	<code>np.array()</code>
Dimensions	<code>ndarray.ndim</code>
Shape	<code>ndarray.shape</code>
Data type	<code>ndarray.dtype</code> , <code>astype()</code>
Size	<code>ndarray.size</code>

---

## Hands-On Practice

- Create 1D, 2D arrays
  - Convert Python list → NumPy array
  - Change datatype (int → float)
- 

## Mini-Project (Real-World)

### Student Marks Analyzer

- Store marks of students
  - Find total subjects, datatype, dimensions
  - Convert marks to float for calculations
- 

## Common Mistakes & Best Practices

- ✗ Assuming NumPy arrays behave like lists
  - ✗ Ignoring dtype (causes bugs later)
  - ✓ Always check shape before operations
  - ✓ Prefer vectorized ops over loops
- 

## Revision Checklist

- What is ndarray?
- List vs NumPy array difference
- What is dtype?
- What is vectorization?

# Week 2 – Array Creation, Indexing, Slicing & Reshaping

## Weekly Goal

Master accessing and restructuring NumPy data.

---

## Key Concepts & Theory

- Array creation methods
  - Indexing (1D, 2D)
  - Slicing rules
  - Reshaping arrays
  - Views vs copies (important!)
- 

## Important Functions & Methods

### Purpose

### Functions

Create arrays

`zeros()`, `ones()`, `arange()`, `linspace()`

Reshape

`reshape()`, `ravel()`, `flatten()`

Transpose

`T`

Copy

`copy()`

---

## Hands-On Practice

- Slice rows & columns
  - Convert 1D → 2D
  - Flatten matrix
  - Test view vs copy behavior
- 

## Mini-Project

### Image Pixel Matrix Simulator

- Create 28×28 grayscale image
  - Extract central region
  - Flatten image for ML model input
- 

## Common Mistakes

- ✗ Confusing `flatten()` vs `ravel()`
  - ✗ Indexing rows incorrectly in 2D arrays
  - ✓ Use `reshape(-1, n)` smartly
- 

## Revision Checklist

- Indexing syntax
- Difference: view vs copy
- Reshape rules

# Week 3 – Math Operations, Broadcasting & Aggregation

## Weekly Goal

Perform fast mathematical operations without loops.

---

## Key Concepts

- Element-wise operations
  - Broadcasting rules
  - Aggregation functions
  - Axis concept
- 

## Important Functions

### Category

### Functions

Math

`add()`, `subtract()`, `multiply()`, `divide()`

Aggregation

`sum()`, `mean()`, `std()`, `min()`, `max()`

Axis

`axis=0`, `axis=1`

---

## Hands-On Practice

- Normalize data
- Row-wise vs column-wise sums

## Mini-Project

### Sales Performance Analyzer

- Daily sales matrix
  - Total sales per product
  - Average sales per day
- 

## Common Mistakes

- ✗ Misusing `axis`
  - ✗ Looping instead of vectorization
  - ✓ Always visualize shape before ops
- 

## Revision Checklist

- Broadcasting rules
- Axis usage
- Vectorized math

# Week 4 – Advanced Indexing, Boolean Masking & Vectorization

## Weekly Goal

Filter and transform data like a pro.

---

## Key Concepts

- Boolean indexing
  - Fancy indexing
  - Conditional selection
  - Vectorization vs loops
- 

## Important Functions

### Purpose

### Functions

Filtering

```
arr[arr > x]
```

Conditions

```
where()
```

Replace

```
clip()
```

---

## Hands-On Practice

- Filter negative values
  - Replace outliers
  - Select specific rows/columns
- 

## Mini-Project

### Sensor Data Cleaning

- Remove faulty readings
  - Cap extreme values
  - Extract valid signals
- 

## Best Practices

- ✓ Prefer boolean masking
  - ✓ Avoid Python `for` loops
  - ✗ Modifying arrays without copy awareness
- 

## Revision Checklist

- Boolean indexing
- Vectorization advantage
- `where()` usage

# Week 5 – Linear Algebra, Statistics & Random Module

## Weekly Goal

Use NumPy for ML math & statistics.

---

## Key Concepts

- Vectors & matrices
  - Dot product
  - Statistical operations
  - Random data generation
- 

## Important Functions

Category	Functions
Linear algebra	<code>dot()</code> , <code>matmul()</code> , <code>linalg.inv()</code>
Stats	<code>mean()</code> , <code>var()</code> , <code>percentile()</code>
Random	<code>rand()</code> , <code>randn()</code> , <code>seed()</code>

---

## Mini-Project

### Simple Recommendation Engine (Math Part)

- User-item matrix
  - Similarity via dot product
- 

### Interview Tip

- “Explain dot product and its ML use”
  - Be ready with vector similarity explanation.
- 

### Revision Checklist

- Dot vs matmul
  - Random reproducibility
  - Statistical summaries
-

# Week 6 – Performance, Memory & Universal Functions (ufuncs)

## Weekly Goal

Write **fast & memory-efficient** NumPy code.

---

## Key Concepts

- Time complexity
  - Memory layout
  - ufuncs
  - In-place operations
- 

## Important Functions

Purpose	Functions
Speed	<code>vectorize()</code>
Memory	<code>copy()</code> , <code>view()</code>
ufuncs	<code>np.sin()</code> , <code>np.exp()</code>

---

## Mini-Project

### Stock Price Simulation

- Generate prices
  - Apply transformations
  - Optimize speed
- 

### Best Practices

- Use ufuncs
  - Avoid unnecessary copies
  - Benchmark code
- 

### Revision Checklist

- ufunc benefits
  - Memory optimization
  - Performance tips
-

# Week 7 – NumPy with Real Data & Pandas Integration

## Weekly Goal

Use NumPy in real data workflows.

---

## Key Concepts

- Loading CSV data
  - Cleaning with NumPy
  - NumPy + Pandas workflow
- 

## Important Functions

Purpose	Functions
Load data	<code>loadtxt()</code> , <code>genfromtxt()</code>
Missing values	<code>nan</code> , <code>nanmean()</code>
Pandas bridge	<code>.values</code> , <code>.to_numpy()</code>

---

## Practice Datasets

- Kaggle: Sales data
- UCI ML datasets, CSV financial data

## Mini-Project

### Data Cleaning Pipeline

- Load messy CSV
  - Handle missing values
  - Prepare for ML
- 

### Revision Checklist

- NumPy ↔ Pandas
  - Missing values handling
  - Real dataset flow
-

# Week 8 – Capstone Project & Interview-Focused Revision

## Weekly Goal

Become **job-ready**.

---

## Capstone Project (Choose One)

1. **End-to-End Sales Analytics System**
2. **Weather Data Statistical Analysis**
3. **ML Data Preprocessing Engine**
4. **Image Data Normalization Tool**

### Must Include

- Data loading
  - Cleaning
  - Aggregation
  - Vectorization
  - Performance optimization
-

# Interview Preparation

## Common Questions

- Why NumPy faster than lists?
- Broadcasting explained
- View vs copy
- Axis logic

## Coding Tasks

- Normalize dataset
  - Filter outliers
  - Compute statistics
- 

## Final Revision Checklist

- ndarray mastery
  - Broadcasting
  - Vectorization
  - Real-world usage
  - Interview confidence
-

## 🎓 Final Outcome (2026-Ready)

By the end of this roadmap, you can:

- Write **production-level NumPy** code
- Prepare data for ML pipelines
- Optimize performance
- Clear **NumPy** interview rounds
- Confidently move to **Pandas, ML & Deep Learning**

---

👉 Follow My Tech [blog](#)

👉 Visit [TechSkillForge](#)

👉 Follow [@pythonquizhub](#) on Instagram

👉 Follow [GitHub](#) | [LinkedIn](#)