Python Libraries Refresher—AI/ML

Python's Role in AI/ML Ecosystem

Data Science Storytelling

Jupyter Notebooks enable data science storytelling.

Simplifies Experimentation

Python's ecosystem simplifies experimentation

Community Support

Python's community supports research and open-source projects.

Accelerates Innovation

Accelerates innovation through dynamic typing and clean syntax.

Flexible Experimentation

Monkey-patching allows flexible experimentation.

Supports AI Pipeline

Python supports every stage of the AI pipeline.

High-Speed Computation

Libraries like NumPy provide high-speed computation.

Discovering The Zen of Python

What happens when I type `import this` in Python?

You will discover "The Zen of Python".

```
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>>
```

Python Data Science Ecosystem

Machine learning tools

Enhanced statistical plots

Basic plotting library

Data manipulation with Data Frames

Foundation for numerical computing

Scikit-learn

Seaborn

Matplotlib

Pandas

NumPy
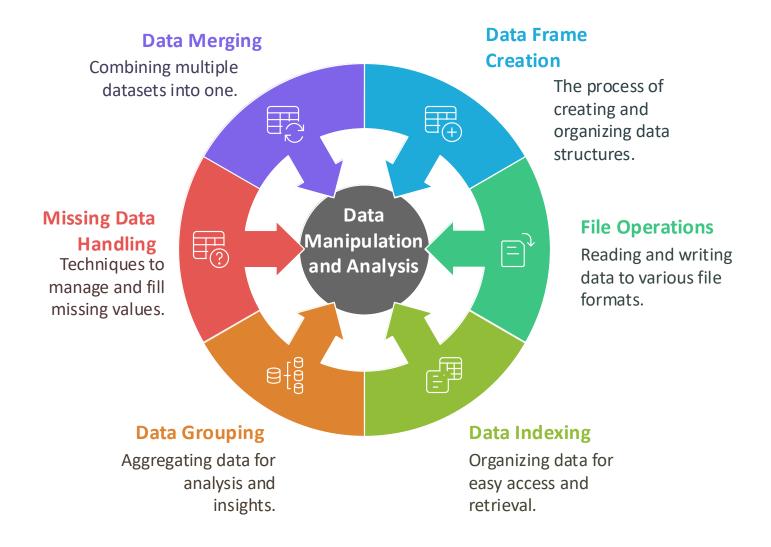
Core Functionalities of NumPy

Reshaping

Changing the dimensions of an array without altering data.

3

Broadcasting

Expanding arrays to perform arithmetic operations.

4

Indexing & Slicing

Techniques to access and modify array elements.

2

NumPy's Core Functionalities

Vectorized Operations

Executing operations on entire arrays efficiently.

5

Array Creation

The process of generating multi-dimensional arrays.

1

Common Functions

Essential mathematical functions for array manipulation.

6

Key Features of Pandas

Data Merging
Combining multiple datasets into one.

Data Frame Creation
The process of creating and organizing data structures.

Missing Data Handling
Techniques to manage and fill missing values.

File Operations
Reading and writing data to various file formats.

Data Manipulation and Analysis

Data Grouping
Aggregating data for analysis and insights.

Data Indexing
Organizing data for easy access and retrieval.

Matplotlib and Seaborn

Line Plots

Used for showing trends over time

Histograms

Effective for showing data distribution

Bar Plots

Useful for comparing quantities across categories
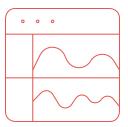
Boxplots

Provides summary statistics of data

Scatter Plots
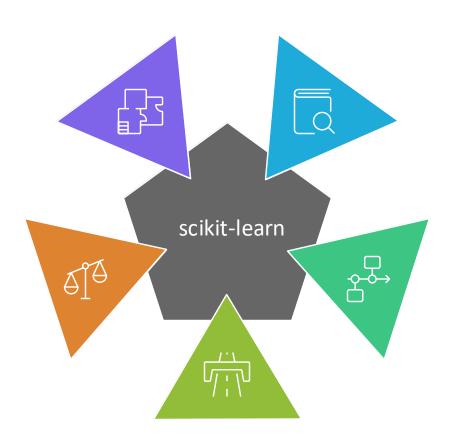
Ideal for displaying relationships between variables

Customizing Plots

Involves adding titles, labels, and legends

Scikit-Learns Features

Machine Learning Toolkit

Provides tools for data mining and analysis

Algorithms

Offers a wide range of ML algorithms

Consistent API

Ensures a uniform interface for models

Model Evaluation

Includes tools for dataset splitting and metrics

Integration

Seamlessly connects with other libraries

scikit-learn

Library Comparison

Library	Core Computing	Data Manipulation	Visualization	Machine Learning
NumPy	High-performance	None	None	None
Pandas	Built on NumPy	Powerful cleaning	None	None
Matplotlib	Works with NumPy	None	Granular control	None
Seaborn	Built on Matplotlib	None	Smart defaults	None
Scikit-learn	Built on NumPy	None	Works with visualizations	Full suite

Streamlit: Fast, Interactive, Intuitive, Visual.

- 📊 **Instant Data Visualization**
Effortlessly display Pandas Data Frames, Matplotlib/Seaborn plots and charts.

- 🚦 **Interactive Model Tuning**
Add sliders, dropdowns, and text inputs to adjust model parameters on-the-fly.

- 📦 **Model Deployment Made Simple**
Share and test ML models without Flask, Django, or full-stack knowledge.

- 🧠 **Explainable AI Demos**
Build apps to visualize SHAP values, LIME outputs, and feature importances interactively.

- 🔁 **Reproducible ML Workflows**
Create end-to-end demos: data loading → preprocessing → training → evaluation → inference.

Thank you