Project Overview: Introduction to Python Programming for Data Science

December 30, 2024

1. Introduction

This project aims to deliver a 15-session course titled **"Introduction to Python Programming for Data Science."** It will guide participants with minimal coding experience through fundamental Python skills, Object-Oriented Programming (OOP), API integration, and essential data science libraries (`NumPy`, `pandas`, `matplotlib`).

Key Goals:

- Build a strong foundation in Python syntax and core concepts.
- Teach OOP principles for structured code design.
- Demonstrate how to fetch and parse real data via APIs and JSON.
- Practice essential data analysis workflows using `NumPy` and `pandas`.
- Introduce basic data visualization with `matplotlib`.
- Culminate with a final project that integrates all learned skills.

2. Intermediate Milestones

Smaller Projects (1.5–2 hours each):

- **OOP Project: Library Management System**
  Design classes (e.g., `Book`, `Library`) to manage book records, demonstrate inheritance and error handling.
- **API-to-File Pipeline:**
  Fetch real data using `requests`, parse JSON, save results to CSV/JSON, handle potential connection errors.
- **Data Cleaning & Exploration:**
  Use `pandas` to clean a small dataset (handle missing values, basic stats, simple visualization).

S#	Topic	Main Focus
1	*Python Basics*	Basic syntax, environment setup, data types.
2	*Control Flow*	Conditionals, loops (for/while), logical operators.
3	*Lists, Tuples*	Data structures, slicing, list methods.
4	*Dictionaries, Sets*	Key-value pairs, set operations, nested structures.
5	*Functions (Basics)*	Defining functions, parameters, return values.
6	*Adv. Functions + Error Handling*	Recursion, lambda, `try/except`, debugging.
7	*OOP Part 1*	Classes, objects, constructors, attributes, methods.
8	*OOP Part 2*	Inheritance, polymorphism, encapsulation.
9	*API Calls & JSON*	Using `requests`, parsing JSON, HTTP basics.
10	*File Handling*	Reading/writing files, storing API data (CSV, JSON).
11	*NumPy Intro*	Arrays vs. lists, vectorized operations, slicing arrays.
12	*pandas Basics*	DataFrames, reading CSV, filtering, cleaning data.
13	*Data Cleaning & Visualization*	Missing data, `matplotlib` basics, plotting.
14	*Integration & Review*	Tying together OOP, API, data analysis for final project.
15	*Final Project Demos*	Student presentations, wrap-up, next steps.

Table 1: Session-by-Session Outline

3. Final Project

A capstone-style *Data Analysis Project*, bringing together:

- **Data Ingestion:** Retrieve data from an API or file source.
- **Data Wrangling:** Clean, transform, and prepare the dataset for analysis using `pandas` and `NumPy`.
- **Visualization:** Present key insights with `matplotlib` charts or plots.
- **OOP and Error Handling:** Showcase class design, plus robust handling of edge cases or invalid inputs.

Students will present their findings in Session 15, demonstrating a clear end-to-end data analysis workflow.

4. Timeline and Deliverables

Week	Topics Covered	Key Deliverables
1–2	Python Basics, Control Flow, Data Structures	Simple Homework Assignments
3–4	Functions, Error Handling, OOP Fundamentals	OOP Project (Library System)
5–6	API/JSON, File I/O	API-to-File Pipeline Project
7–8	NumPy, pandas, Visualization	Data Cleaning & Exploration Project
End of 8	Final Integrations	Final Project Presentation

Table 2: Projected Timeline of Deliverables

Note: Each week comprises two 1.5-hour sessions. The schedule is flexible to accommodate learners' pacing and project progress.

5. Conclusion

By the end of this program, participants will have:

- A thorough understanding of Python's core features and OOP concepts.
- Practical experience calling APIs, parsing JSON, and storing results.
- A solid grasp of essential data science libraries for analysis and visualization.

- Completed multiple hands-on projects, culminating in a final data-driven capstone.

This structured, incremental approach ensures that new learners build confidence while gaining marketable data science skills.