Introduction to Python and Data Science with AI Enhancements using LLMs

Module 1: Detailed Python 3.14

Introduction to Python

- **What is Python?**
  - Overview: High-level programming language with dynamic typing and memory management.
  - Applications: AI, Data Science, and Automation.
- **New Features in Python 3.14**
  - Syntax improvements.
  - Enhanced type hints and error handling.

Setting Up Python Environment

- **Installing Python 3.14**
  - Installation via Anaconda, Python.org, or package managers.
- **Working with IDEs**
  - Introduction to VSCode, PyCharm, Jupyter Notebook.
- **Using Virtual Environments**
  - Tools: `venv`, `pipenv`, `conda environments`.
  - Managing dependencies efficiently.

Core Python Syntax and Concepts

- **Variables and Data Types**
  - Examples: Integers, Floats, Strings, Booleans, Lists, Tuples, Sets, Dictionaries.
- **Basic Operators**
  - Arithmetic, Comparison, Logical, and Bitwise Operators.
- **Control Flow**
  - Conditional Statements: `if`, `elif`, `else`.
  - Loops: `for`, `while`, `break`, `continue`.
- **Functions and Modules**
  - Defining, returning values, and importing libraries.

Advanced Python Concepts

- **List Comprehensions and Generator Expressions**
- **Lambda Functions and Higher-Order Functions**
  - Tools: `map()`, `filter()`, `reduce()`.
- **Error Handling**
  - `try`, `except`, `finally` blocks; Custom exception handling.
- **Working with Files**

    - Operations: Reading/writing text, binary files, CSV, and JSON.
- **Python Best Practices**
    - PEP 8 guidelines, debugging (`pdb`), and testing (`unittest`, `pytest`).

Module 2: NumPy for AI

Introduction to NumPy

- **What is NumPy?**
    - Essential library for numerical computing.
    - Efficient multidimensional array manipulation.
- **Installation and Import**
    - `pip install numpy`.
    - `import numpy as np`.

NumPy Arrays

- **Creating Arrays**
    - Tools: `np.array()`, `np.zeros()`, `np.ones()`, `np.arange()`, `np.linspace()`.
- **Array Dimensions and Indexing**
    - Shape, size, slicing, and boolean indexing.

Array Operations

- **Element-wise Operations**
    - Arithmetic operations and universal functions (`ufuncs`).
- **Linear Algebra**
    - Matrix operations (`np.dot()`), Eigenvalues, SVD.
- **Statistical and Mathematical Functions**
    - Mean, median, variance, `np.exp()`, `np.log()`.

Advanced Techniques

- **Performance Optimization**
    - Using arrays over Python lists.
- **Practical Exercises with NumPy**
    - Solving mathematical problems using arrays
    - Performing matrix manipulations and linear algebra

Module 3: Pandas for Data Manipulation

Introduction to Pandas

- **What is Pandas?**

    - Structured data handling.
    - Built on NumPy.

Pandas Data Structures

- **Series and DataFrames**
    - Creating and manipulating data.
    - Indexing, slicing, and filtering.

Data Cleaning and Preprocessing

- **Handling Missing Data**
    - Identifying, filling, and dropping missing values.
- **Data Transformation**
    - Renaming columns, type conversion, normalization.

Data Aggregation

- **Grouping and Pivot Tables**
    - Tools: `groupby()`, aggregation functions.

Time Series Analysis

- **Datetime Operations**
    - Resampling and shifting data.

Practical Exercises with Pandas

- Data exploration and cleaning on real-world datasets
- Data analysis tasks like filtering, grouping, and summarizing data
- Visualizing data with Pandas and integration with Matplotlib

Module 4: Data Visualization with Matplotlib

Introduction to Matplotlib

- **What is Matplotlib?**
    - Static and interactive visualizations.
- **Installation and Import**
    - `pip install matplotlib`.
    - `import matplotlib.pyplot as plt`.

Creating Basic Plots

- Line, Bar, Scatter, and Histogram plots.
- Customizing titles, labels, and grids.

Advanced Visualization Techniques

- **Subplots and Pie Charts**
- **Box Plots and Heatmaps**
- **Saving Plots**
  - Formats: PNG, PDF.

Practical Exercises

- Visualizing data from Pandas DataFrame
- Creating various types of plots based on dataset features
- Combining plots for better insights

Final Project: Integrating Python, NumPy, Pandas, and Matplotlib

Leveraging LLMs for Enhanced Learning

- **Code Assistance**
  - Generating boilerplate code and debugging with LLMs.
- **Learning Support**
  - Explaining concepts, suggesting optimizations, and answering queries.
- **Case Studies**
  - Interactive Q&A to solve real-world problems.