

# Data Science Roadmap - Week 1

## Week 1: Introduction to Data Science and Python Basics

### 1. Introduction to Data Science

#### - Overview of Data Science

- Definition and Importance: Understand what data science is and why it's important in modern decision-making.

- Applications: Explore real-world applications (e.g., recommendation systems, image recognition, fraud detection).

- Comparison with Related Fields: Differentiate between data science, data analytics, machine learning, and artificial intelligence.

#### - Roles in Data Science

- Data Scientist: Key skills (statistical analysis, machine learning, programming).

- Data Analyst: Emphasis on visualization, dashboards, and business insights.

- Data Engineer: Data pipeline design, ETL processes, and data warehousing.

- Machine Learning Engineer: Focus on model building, deployment, and scalability.

#### - Data Science Tools and Ecosystem

- Programming Languages: Overview of Python, R, and SQL for data-related tasks.

- Libraries and Frameworks: Introduction to essential libraries like NumPy, Pandas, Matplotlib, and Scikit-Learn.

- Development Environments: Introduction to Jupyter Notebooks, Google Colab, and IDEs like PyCharm.

## - Typical Data Science Workflow

- Problem Definition: Define the business problem and goals.
- Data Collection: Methods for gathering data (APIs, web scraping, databases).
- Data Wrangling: Data cleaning, handling missing values, and formatting.
- Exploratory Data Analysis (EDA): Summarize main characteristics of data using visualization.
- Model Building: Supervised vs. unsupervised models, training, and testing.
- Evaluation and Interpretation: Analyzing model performance and drawing insights.

## 2. Python Basics

### - Setting Up Python Environment

- Installing Anaconda: Guide to setting up Anaconda for a seamless Python experience.
- Introduction to Jupyter Notebooks: Writing and executing Python code interactively.

### - Basic Python Syntax

- Data Types: Understanding and using integers, floats, strings, booleans, and complex numbers.
- Variables and Naming Conventions: Declaring variables and adhering to naming best practices.
- Basic I/O Operations: Using `input()` for data input and `print()` for output.

### - Control Structures

- Conditionals (`if`, `elif`, `else`): Logical branching for decision-making in code.
- Loops (`for`, `while`): Iterating over data structures; differences between `for` and `while` loops.
- List Comprehensions: Concise way to create lists and perform transformations.

### - Functions in Python

- Defining Functions: Syntax and structure of a function, return statements, and scope.

- Parameters and Arguments: Positional, keyword, and default arguments.
- Lambda Functions: Usage of anonymous functions for short operations.
- Working with Libraries (NumPy, Pandas)
  - NumPy Basics: Working with arrays, basic operations, and reshaping data.
  - Pandas Basics: Creating data frames, basic data operations, and accessing data by indexing.