Python Programming and Basics of machine learning

- What is Python?
- Interpreted, High-Level, and Easy to Learn

History and Evolution

- Guido van Rossum created Python in 1991
- Python 2 vs Python 3

Features of Python

- Easy to Read and Write
- Extensive Libraries
- Portability

Applications of Python

Web Development

Data Science

AI and Machine Learning

Why Choose Python?

- Large Community Support
- Rapid Prototyping and Scalability

Installing Python

- Download from python.org
- Anaconda for Data Science

Choosing IDEs

- Jupyter Notebook
- Spider
- IDLE
- VS Code
- PyCharm

Running Python Code

- Command Line
- Using IDEs
- Interactive Mode

Variables and Data Types

•- int: Integer values
•- float: Decimal values
•- complex: Complex numbers
•- bool: Boolean values
•- str: Text data

cont........

- - list: Mutable sequence
- - tuple: Immutable sequence
- - dict: Collection of key-value pairs

Operators and Expressions

Arithmetic, Logical, Relational Operators

Arithmetic

- Addition (+)
- Subtraction (-)
- Multiplication (*)
- Division (/)
- Floor Division (//)
- Modulus (%)
- Exponentiation (**)

Relational (Comparison) Operators

- Equal to (==)
- Not equal to (!=)
- Greater than (>)
- Less than (<)
- Greater than or equal to (>=)
- Less than or equal to (<=)

Logical Operators

- AND (&)
- OR (|)
- NOT (!)

Control Flow

- if, else, elif Statements
- for and while Loops

Functions and Modules

- Functions

  Def-A **function** is a reusable block of code

- Types

  User Defined

  System Defined

Exception Handling

- try, except, finally
- Handling Errors Gracefully

NumPy for Array Manipulation

- Manipulating Arrays
- Array Operations

Pandas for Data Manipulation

- DataFrames and Series
- Data Cleaning and Transformation

Matplotlib & Seaborn for Visualization

- Line Charts, Bar Graphs
- Heatmaps and Pairplots

What is Machine Learning?

Enabling Computers to Learn from Data

Types of Machine Learning

Supervised, Unsupervised, Reinforcement Learning

Applications of ML

- Spam Detection
- Recommendation Systems

Data Collection

- Gathering Data from Various Sources

Data Preprocessing

- Cleaning and Encoding Data

Model Building

- Choosing Algorithm
- Training the Model

Model Evaluation

- Accuracy, Precision, Recall, F1-Score

Model Deployment

- Deploying Model using Flask/Streamlit

Supervised Learning

- Regression and Classification Models

Unsupervised Learning

- Clustering and Dimensionality Reduction

Scikit-Learn Overview

- Preprocessing, Model Training, Evaluation

TensorFlow Overview

- Deep Learning Framework
- Building Neural Networks

Keras Overview

- High-Level API for Deep Learning

Training a Model

- Splitting Dataset, Model Training

Evaluating and Testing

- Confusion Matrix

Limitations of Programming Scripting Languages for Machine Learning

- Importance of selecting the right language for ML projects
- Scope: Comparison of Python, R, Java, C++ and others.

Python

Advantages:

- Rich ML libraries (ScikitLearn, TensorFlow, PyTorch).
- Easy syntax and large community.

Limitations:

- Slower execution for largescale applications.
- High memory consumption.
- Not ideal for mobile or lowlatency applications.

R

Advantages:

- Statistical analysis and visualization
- Great for data exploration

Limitations:

- Slower in production environments
- Limited deep learning support

Java

Advantages:

- Fast execution and scalability

Limitations:

- Limited ML libraries compared to Python.
- Complex syntax.

C++

Advantages:

- High performance
- Suitable for real time systems.

Limitations:

- Complex memory management.
- Difficult to debug and maintain.
- Slower development cycle.

MATLAB

Advantages:

- Strong in numerical and matrix computations
- Useful for prototyping

Limitations:

- Expensive licenses.
- Limited integration with modern ML frameworks

Introduction to OOP

What is OOP?

- Object Oriented Programming (OOP) is a paradigm based on objects and classes.

Importance and Applications

- Widely used in software development
- reusable, and maintainable code
- Ideal for complex systems like web applications and AI

Key Principles Overview of OOPS

- Abstraction

Def- Hides unnecessary details and provides only essential information to the user

Ex. Coffee Machine

cont.....

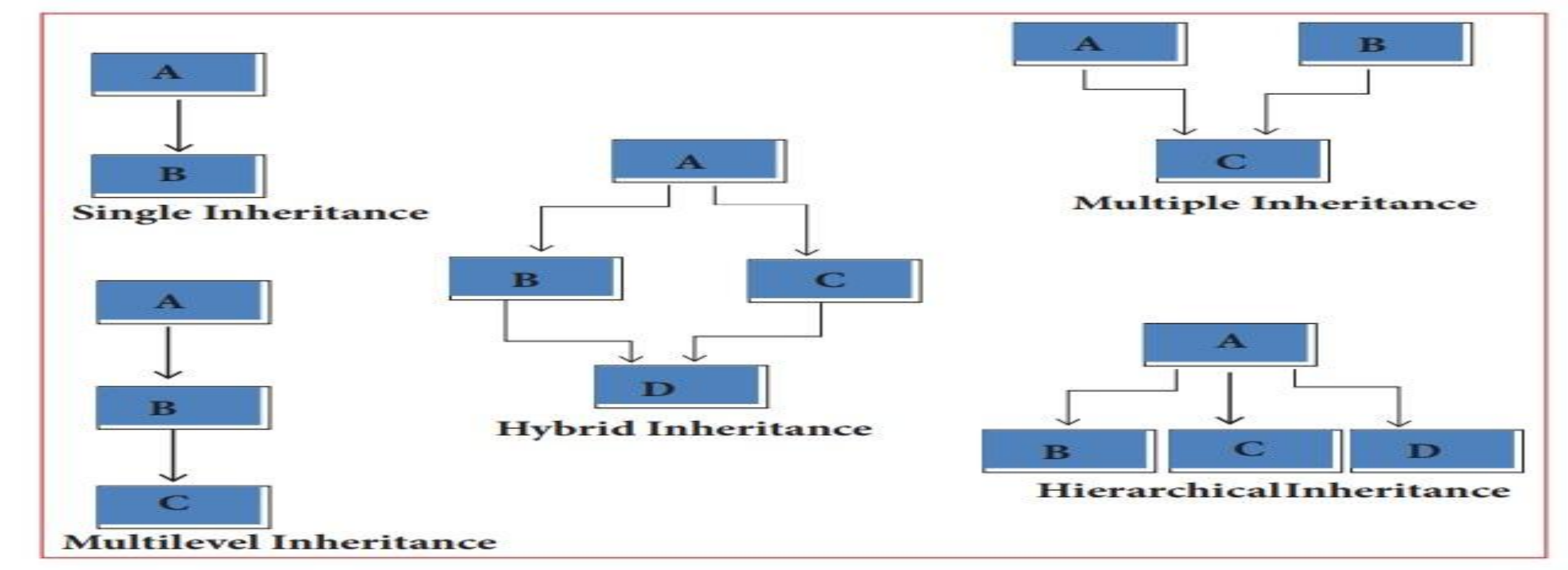
- Encapsulation

Def-Wraps data (variables) and methods (functions) into a single unit (class)

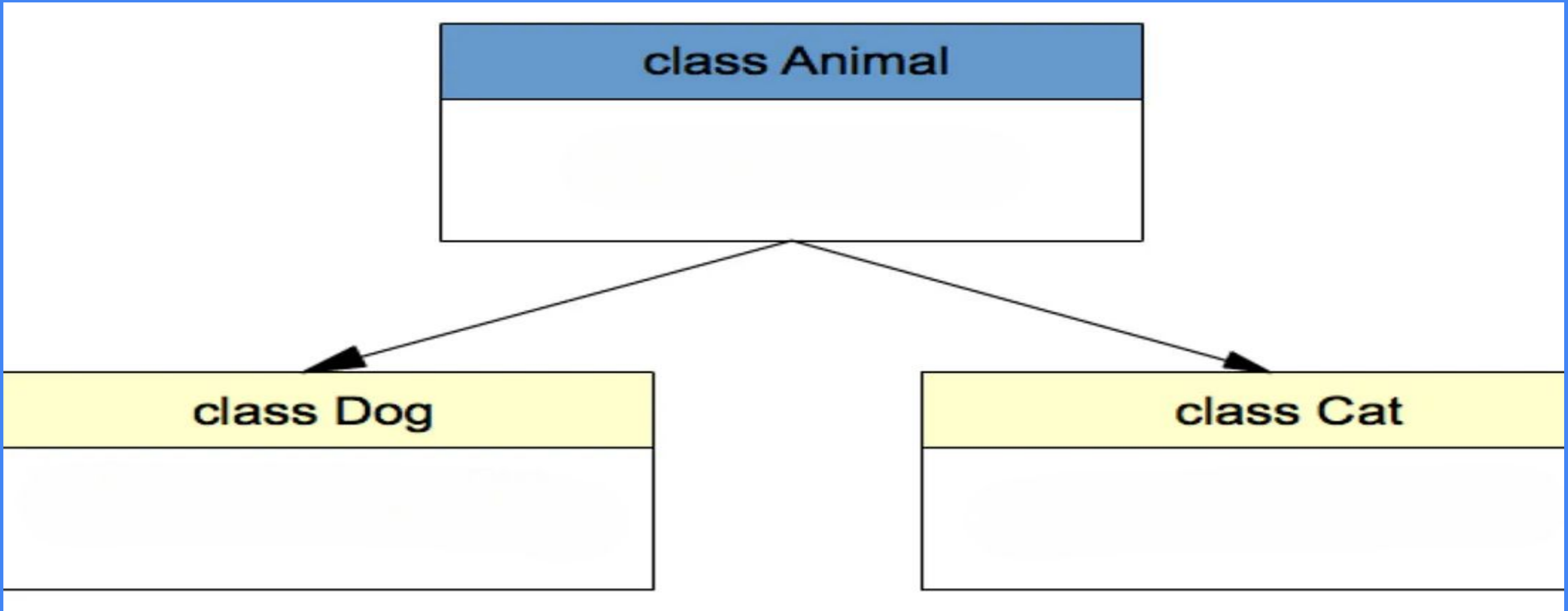
Ex. Bank Account

cont.....

- Inheritance



- Polymorphism



Why Abstraction?

- Reduces complexity
- Promotes security
- Improves maintainability

Public, Protected, Private

- Public: Accessible from anywhere
- Protected: Accessible in subclass
- Private: Accessible within the class