

Week 1 Assignment Report

1. Difference between AI, Machine Learning, Deep Learning, and Data Science

Artificial Intelligence (AI) is the broader field of creating machines that can perform tasks that usually require human intelligence. For example, virtual assistants like Siri and Alexa use AI to process speech and provide responses.

Machine Learning (ML) is a subset of AI where systems learn patterns from data without being explicitly programmed. For example, spam email detection learns from past labeled data to identify spam.

Deep Learning (DL) is a subset of ML that uses neural networks with many layers to model complex patterns. For example, self-driving cars use deep learning for image recognition to detect pedestrians and traffic signs.

Data Science involves extracting insights from data using statistics, ML, and visualization techniques. For example, e-commerce companies use data science to analyze customer purchase patterns and recommend products.

2. Mutable vs Immutable Data Types

Mutable data types can be changed after they are created. For example, lists and dictionaries in Python allow adding, removing, or updating elements. Immutable data types cannot be changed once created. For example, strings and tuples cannot be modified; instead, new objects are created when changes are made.

Example: If you change a list element, the original list is updated. But if you try to modify a string, a new string object is created.

3. Deep Copy vs Shallow Copy

A shallow copy creates a new object but references the elements of the original object. Changes in nested objects affect both copies. For example, using the 'copy' module's `copy()` function.

A deep copy creates a completely independent copy, including nested objects. Changes in one copy do not affect the other. For example, using the 'copy' module's `deepcopy()` function.

In short, shallow copy shares inner references, while deep copy duplicates everything.

4. Git Branching and its Importance

Git branching allows developers to create separate versions of a project to work on features, bug fixes, or experiments without affecting the main codebase. This is crucial in collaborative development because multiple developers can work simultaneously without interfering with each other's work.

For example, a developer can create a 'feature-login' branch to add a login system while others continue improving the UI on another branch. Once tested, the branches can be merged back into the main branch.

End of Report