**1. Introduction to Version Control with Git and GitHub**

**What is Version Control?**

Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later. It helps manage and track changes in source code during software development.

**Git Overview**

**Git:** A distributed version control system that allows multiple developers to work on a project simultaneously without overwriting each other's work.

**Repositories (Repos):** A storage space where your project lives. It can be local to your computer or stored on a remote server (e.g., GitHub).

**Commits:** Snapshots of your project at a point in time. Each commit has a unique ID.

**Branches:** Parallel versions of your repository. You can work on different features without affecting the main project.

**GitHub Overview**

**GitHub:** A web-based platform for version control using Git. It provides a graphical interface, collaboration features, and other tools to facilitate software development.

**Repositories on GitHub:** Public or private storage spaces for your projects.

**Forking:** Creating a personal copy of someone else’s project.

**Pull Requests:** Proposed changes to a repository submitted by a user and accepted or rejected by repository owners.

**2. Git Branching Hands-on Learning**

**Creating and Managing Branches**

**Creating a Branch:** Use git branch [branch-name] to create a new branch.

**Switching Branches:** Use git checkout [branch-name] to switch to the branch.

**Creating and Switching in One Step:** Use git checkout -b [branch-name].

**Merging Branches:** Use git merge [branch-name] to merge changes from one branch into another.

**Deleting a Branch:** Use git branch -d [branch-name] to delete a branch.

Hands-on Practice

**Scenario:** Creating a new feature in a project.

**Create a new branch for the feature:** git checkout -b feature-branch.

**Make changes and commit them:** git add . followed by git commit -m "Added new feature".

**Switch back to the main branch:** git checkout main.

**Merge the feature branch into the main branch:** git merge feature-branch.

**Push changes to the remote repository:** git push origin main.

**3. Understanding Key Terminologies and Differences**

**Artificial Intelligence (AI)**

**Definition:** The broad field of creating intelligent agents capable of performing tasks that typically require human intelligence.

**Examples:** Natural language processing, image recognition, and game playing.

**Machine Learning (ML)**

**Definition:** A subset of AI that involves the development of algorithms that allow computers to learn from and make predictions or decisions based on data.

**Examples:** Spam email detection, recommendation systems, and predictive maintenance.

**Deep Learning (DL)**

**Definition:** A subset of machine learning that uses neural networks with many layers (deep neural networks) to analyze various factors of data.

**Examples:** Speech recognition, image classification, and autonomous driving.

**Data Science**

**Definition:** An interdisciplinary field that uses scientific methods, processes, algorithms, and systems to extract knowledge and insights from structured and unstructureddata**.**

**Examples:** Data mining, statistical analysis, and big data analytics.

**Summary of Learnings**

This week, I gained a fundamental understanding of version control with Git and GitHub, including creating and managing branches to handle different features in a project. Additionally, I learned the key differences and definitions of AI, ML, DL, and Data Science, which are crucial for understanding the scope and applications of each field.