**WEEK 01**

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

**What is Version Control?**

Version control is a system that tracks changes to files over time, allowing you to recall specific versions later. It is essential for managing and tracking changes in source code during software development.

**Git Overview**

* **Git**: A distributed version control system that enables multiple developers to work on a project simultaneously without overwriting each other's work.
* **Repositories (Repos)**: Storage spaces where your project files reside. Repositories can be local on your computer or hosted on a remote server (e.g., GitHub).
* **Commits**: Snapshots of your project at specific points in time. Each commit has a unique ID.
* **Branches**: Parallel versions of your repository that allow you to work on different features without affecting the main project.

**GitHub Overview**

* **GitHub**: A web-based platform for version control using Git. It offers a graphical interface, collaboration features, and additional 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, which can be accepted or rejected by the 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 desired branch.
* **Creating and Switching in One Step**: Use git checkout -b [branch-name] to create and switch to a new branch in one command.
* **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 .

git commit -m "Added new feature"

 **Switch back to the main branch**:

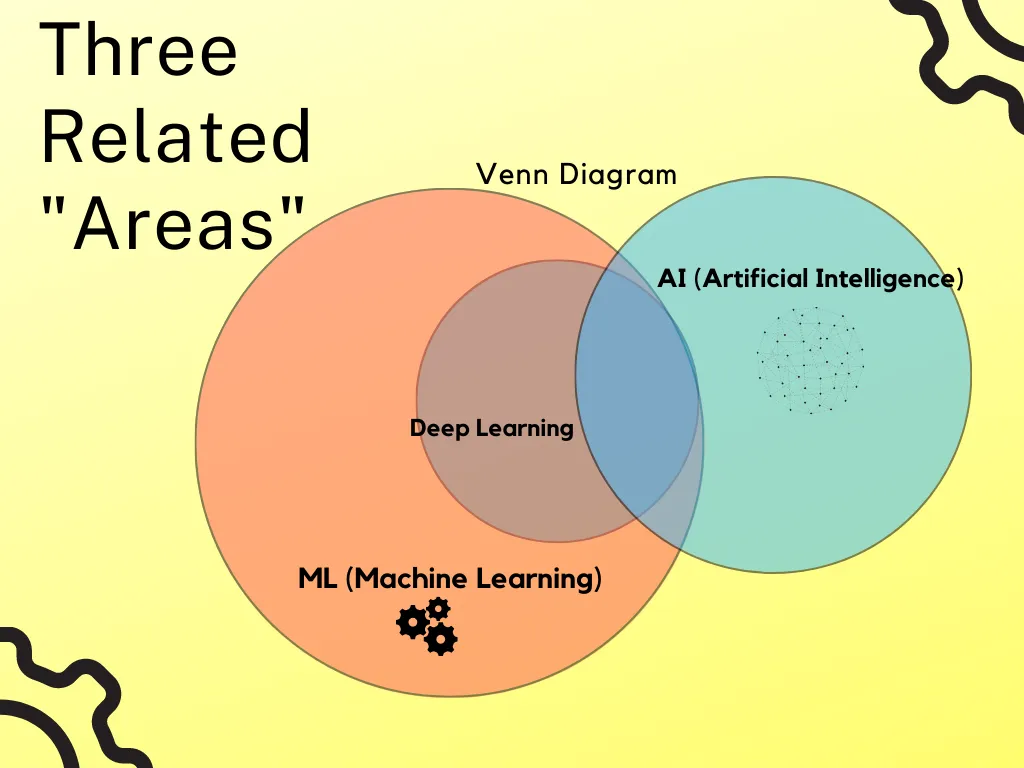
git checkout main

 **Merge the feature branch into the main branch**:

git merge feature-branch

 **Push changes to the remote repository**:

git push origin main



**Artificial Intelligence (AI)**

AI is about making computers smart so they can do things like understand language, recognize images, and make decisions.

* **Examples**: Siri, self-driving cars, Netflix recommendations.

**Machine Learning (ML)**

ML is a way to teach computers to learn from data and improve over time.

* **Examples**: Email spam filters, movie recommendations, predicting house prices.

**Deep Learning (DL)**

DL is a special type of ML that uses complex networks to learn from lots of data.

* **Examples**: Voice assistants, facial recognition, translating languages.

**Data Science (DS)**

DS is about collecting, analyzing, and making sense of data to find useful information.

* **Examples**: Analyzing sales data, studying customer behavior, predicting trends.

**Relationships**

* **AI**: Making computers smart.
* **ML**: Teaching computers to learn from data.
* **DL**: Using advanced ML for better learning.
* **DS**: Finding insights from data using various methods, including ML and DL.