# CI/CD Pipeline for a Machine Learning Project

## Total Marks: 100

## Deadline: March 07, 2025

**Description:**

This assignment aims to provide hands-on experience in designing and implementing a **CI/CD pipeline** for a **machine learning project**. The project will include both a **machine-learning model** and a **unique dataset** for each group.

**Group Formation:**

- This is a **group assignment** with a maximum of **two members per group**.
- Each group must designate **one member as the admin** responsible for reviewing and approving pull requests before merging changes into the repository.

**Required Tools:**

Students must use the following tools while developing the pipeline:

1. **Jenkins**
2. **GitHub**
3. **GitHub Actions**
4. **Git**
5. **Docker**
6. **Python**
7. **Flask**

**Task Breakdown:**

1. **Repository Setup & Branching Strategy**
    - Create a GitHub repository with the following branches:
        - **Dev Branch:** For development and feature implementation.
        - **Test Branch:** For validating features before production.
        - **Master Branch:** For final, production-ready code.
2. **Code Quality Check (GitHub Actions & Flake8)**
    - Implement a **GitHub Actions workflow** to enforce **code quality checks** using **Flake8**.
    - Ensure that any pull request to the **dev branch** must pass this check before merging.
3. **Feature Testing (GitHub Actions)**
    - Once a **feature is completed** in the **dev branch**, submit a **pull request** to merge it into the **test branch**.
    - This should trigger an **automated testing workflow**, executing **unit tests** to validate the feature.
4. **Deployment with Jenkins & Docker**
    - Upon successful testing, merge the feature into the **master branch**, triggering a **Jenkins job**.
    - The Jenkins job should:
        - **Containerize the application** using **Docker**.
        - **Push the Docker image** to **Docker Hub**.
5. **Admin Notification**
    - Once the merge into the **master branch** is complete, an **email notification** should be sent to the **admin**, confirming the successful deployment via Jenkins.

**Evaluation Criteria:**

- Proper **repository structure** and **branching strategy** (20 Marks)
- Successful **code quality enforcement** using **Flake8** (20 Marks)
- Correct **implementation of unit testing workflow** (20 Marks)
- Functional **Jenkins & Docker integration** for deployment (30 Marks)
- Proper **admin notification setup** (10 Marks)

**Submission Guidelines:**

- Submit the **GitHub repository link** along with a short report explaining your implementation.
- Ensure that your pipeline follows the required structure and meets all conditions mentioned above.

**Best of Luck.**