**MLOps Assignment 1**

**Task M1 Deliverable: Report - CI/CD Pipeline Stages**

**Introduction**

This report outlines the CI/CD pipeline stages implemented for a sample machine learning project using GitHub Actions. The pipeline automates the processes of code validation, model training, and deployment, ensuring a streamlined and efficient workflow from development to production.

**GitHub Repository Link**: [himanshuamodwala/MLOpsAssignment1 (github.com)](https://github.com/himanshuamodwala/MLOpsAssignment1) (Currently Private, will be made Public after the Assignment deadline has passed to avoid plagiarism)

**Continuous Integration (CI) Pipeline**

The CI pipeline is designed to trigger on pull requests to the main branch. It includes the following stages:

|  |  |
| --- | --- |
| Code Linting: | * **Purpose:** To ensure that the Python code adheres to coding standards and is free of syntax errors. * **Tool Used:** flake8 * **Process:**   + The code is checked out from the repository.   + **Python** and **flake8** are set up in the environment.   + The code checked for linting errors. E.g. – flake8 M3/m3.py   + If any linting errors are found, an issue is created in the repository, and the pipeline fails.   + If no errors are found, a comment is added to the pull request indicating successful linting. |
| Model Building: | * **Purpose**: To automate the training of a machine learning model and log the best hyperparameters. * **Tool Used:** Python (with scikit-learn, Optuna, and joblib) * **Process:**   + The code is checked out from the repository.   + **Python** and the necessary dependencies are set up.   + The model training script is executed and the best hyperparameters are extracted.   + A comment is added to the pull request with the best hyperparameters. The model is saved as an artifact for later stages. |
| Unit Testing: | * **Purpose: To ensure the model parameters are within the expected range.** * **Process:**    + **The saved model artifact is retrieved.**   + **The model is loaded, and its hyperparameters (n\_estimators, max\_depth, min\_samples\_split) are validated.**   + **If the parameters are within the expected range, a comment is added to the pull request indicating success.**   + **If the parameters are outside the expected range, an issue is created, and the pipeline fails.** |

**Continuous Deployment (CD) Pipeline**

The CD pipeline is designed to be triggered manually via **workflow\_dispatch**. It includes the following stages:

|  |  |
| --- | --- |
| Branch Validation: | * **Purpose:** To ensure the deployment is only executed from the main branch. * **Process:**   + The branch reference is checked.   + If the branch is not main, the job fails.   + If the branch is main, the workflow proceeds. |
| Model Training: | * **Purpose:** To ensure the latest code changes are included in the model version to be deployed. * **Tool Used:** Python (with scikit-learn, Optuna, and joblib) * **Process:**   + The latest version of the main branch is checked out.   + **Python** and the necessary dependencies are set up.   + The model training script is executed.   + The model is saved as an artifact for deployment. |
| Docker Image Creation and Deployment: | * **Purpose:** To package the trained model into a Docker container and push it to Docker Hub. * **Tool Used:** Docker * **Process:**    + The saved model artifact is retrieved.   + Docker is set up.   + Login to Docker Hub is performed using credentials stored as secrets.   + A Docker image is built using the model and other necessary scripts of the flask application.   + The Docker image is tagged with a timestamp and pushed to Docker Hub as both a timestamped and latest version. |

**Conclusion**

The CI/CD pipelines effectively automate the processes of code validation, model training, and deployment, providing a robust framework for continuous integration and deployment in machine learning projects.

The use of GitHub Actions for CI/CD, along with Docker for containerization, ensures consistency, reliability, and scalability of the application from development to production.

**Task M1 Deliverable: Screenshots showing successful runs of the pipeline**

**The CI pipeline detects the linting errors in the script; logs a GitHub issue; and fails the pipeline.**

|  |  |
| --- | --- |
| Script with linting errors | Creating pull request to merge feature branch (code with linting errors) with the main branch. |
| CI pipeline GitHub Action is auto triggered. | The action detects linting errors, raises a GitHub issue, references it in the PR  A screenshot of a computer  Description automatically generatedA screenshot of a computer error  Description automatically generated |
| The action fails the CI pipeline | A screenshot of a computer  Description automatically generated |

**The CI pipeline detects unit testing failures; logs a GitHub issue; and fails the pipeline.**

|  |  |
| --- | --- |
| Script with no linting errors | Creating pull request to merge feature branch with the main branch. |
| CI pipeline GitHub Action is auto triggered.  **A screenshot of a computer  Description automatically generated** | No linting error found, model was built properly.  A screenshot of a computer  Description automatically generated |
| UT fails as hyperparameters out of acceptable range. GitHub issue is raised and references in the PR.  **A screenshot of a computer  Description automatically generated** | The action fails the CI Pipeline. |

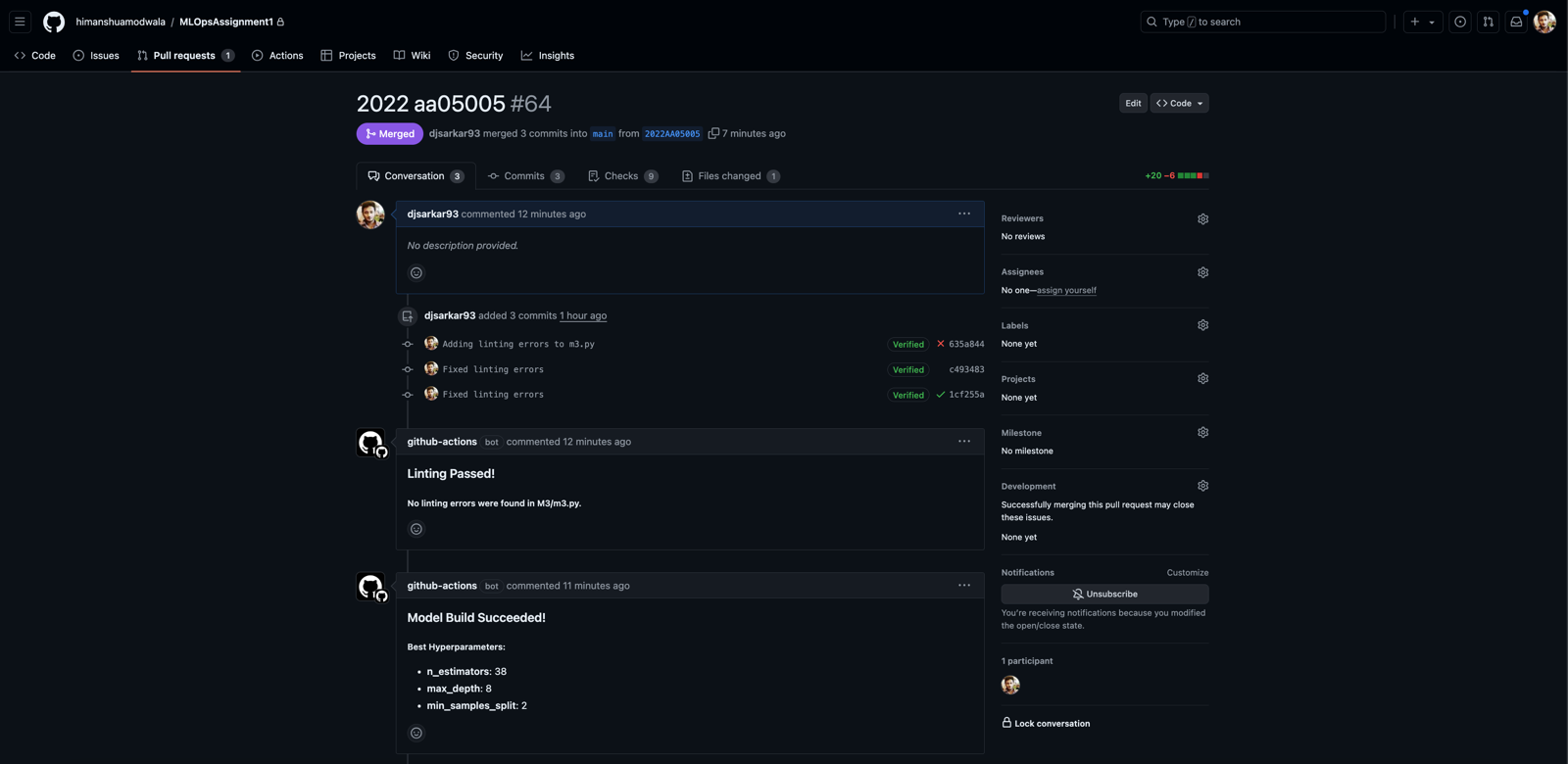
**The CI pipeline completes successfully.**

|  |  |
| --- | --- |
| Script with no linting errors | Creating pull request to merge feature branch with the main branch. |
| CI pipeline GitHub Action is auto triggered.  **A screenshot of a computer  Description automatically generated** | No linting error found, model was built properly, and all unit test cases passed. |
| The CI pipeline completes successfully.  A screenshot of a computer  Description automatically generated |  |

**The CD pipeline completes successfully.**

|  |  |
| --- | --- |
| CD pipeline is triggered manually  **A screenshot of a computer  Description automatically generated** | The model is built and successfully deployed to Docker Hub.  **A screenshot of a computer  Description automatically generated** |
| The CD pipeline completes successfully.  A screenshot of a computer  Description automatically generated |  |

**Task M1 Deliverable: Git branch merge history**

****

**A screenshot of a computer

Description automatically generated**

Git repository screenshot with branches and merge history

**A screenshot of a computer

Description automatically generated**