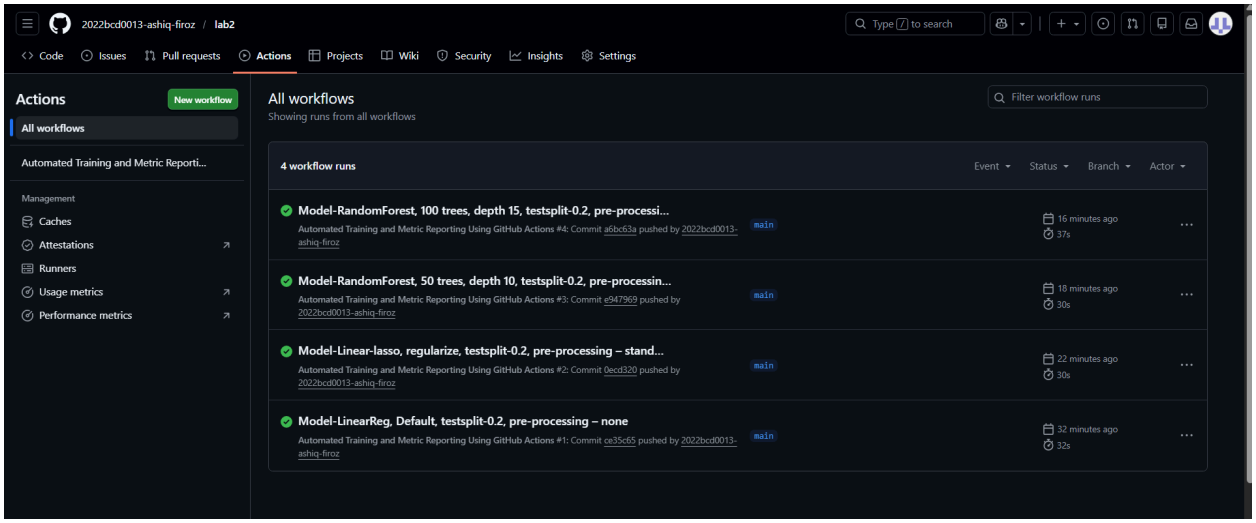
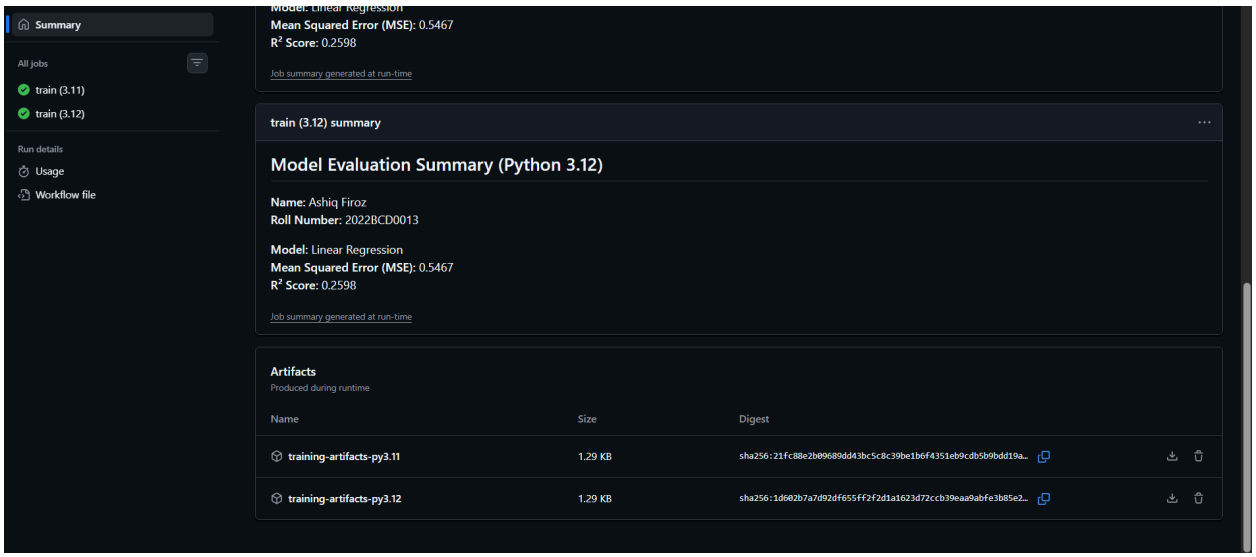
MLOPS Lab 2

Name : Ashiq Firoz
Roll : 2022BCD0013

Github Repository Link : https://github.com/2022bcd0013-ashiq-firoz/lab2

Job summary with metrics for all experiments





1. How did GitHub Actions improve experiment reproducibility?

GitHub Actions enforces a fixed execution environment, dependency versions, and scripted training steps. Every run is automatically triggered and executed identically on clean machines. This eliminates environment drift common in local or notebook-based experiments.

2. How easy was it to compare results across runs?

Comparison was straightforward since metrics were logged in a consistent JSON format and summarized per run. Each workflow execution stored artifacts and evaluation summaries. Python version matrices further enabled systematic cross-environment comparison.

3. What role does Git commit history play in experiment tracking?

Each experiment is tied to a specific commit hash, ensuring code–result traceability. Changes in performance can be directly mapped to code modifications. This creates an implicit, version-controlled experiment log.

4. What were the benefits of this approach compared to Lab 1?

Unlike Lab 1’s manual notebook runs, this approach automated training, evaluation, and logging. It supported multiple models and environments without manual intervention. Results were reproducible, auditable, and centrally stored.

5. What limitations does this approach have?

GitHub Actions has limited compute resources and execution time. It is less interactive than notebooks for exploratory analysis. Advanced experiment management (e.g., hyperparameter sweeps) requires additional tooling.