

CSS 426 - MLOps

Name :- VISHNU NARAYANAN VINODKUMAR

Roll Number :- 2022BCS0001

Date :- 09-01-2026

Batch :- 1

Lab :- 2

Repository Link - https://github.com/2022bcs0001-vishnu/LAB_2.git

Problem

Task 1: Create a GitHub repository

You are requested create a new GitHub account with username in the format <rollno>< name >(preferably using institute email id). Ex: 2022bcs0001_vishnu. Create a new public repository “lab2” and organize your project folder as follows

- Dataset directory
- Training script
- Requirements file
- Output directory for:
 - Trained model
 - Evaluation results (JSON)
- GitHub Actions workflow directory

Output

Account created

☰ 2022bcs0001-vishnu

Overview Repositories Projects Packages Stars

Popular repositories

You don't have any public repositories yet.

Customize your pins

1 contribution in the last year

Contribution settings ▾ 2026

Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

Mon Wed Fri

Learn how we count contributions Less More

This is your **contribution graph**. Your first square is for joining GitHub and you'll earn more as you make additional contributions. More contributions means a higher contrast square color for that day. Over time, your chart might start looking something like this.

We have a quick guide that will show you how to create your first repository and earn more green squares!

Read the Hello World guide

2022bcs0001-vishnu
2022bcs0001-vishnu · he/him

IIIT Kottayam - Lab account Vishnu Narayanan Vinodkumar

Edit profile

IIT Kottayam

Joined 12 minutes ago

☰ Settings

Type / to search

Go to your personal profile

Public profile

Name: 2022bcs0001-vishnu

Your name may appear around GitHub where you contribute or are mentioned. You can remove it at any time.

Profile picture

Public email: vishnu22bcs1@iitkottayam.ac.in

You can manage verified email addresses in your [email settings](#).

Bio: IIIT Kottayam - Lab account
Vishnu Narayanan Vinodkumar

>Edit

Pronouns: he/him

URL:

Social accounts

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and licensing

Emails

Password and authentication

Sessions

SSH and GPG keys

Organizations

Enterprises

Moderation

Code, planning, and automation

Repositories

Codespaces

Search

ENG IN

10:23 AM 09-01-2026

Repository named LAB_2 has been made

The screenshot shows a GitHub repository page for a public repository named 'LAB_2'. The repository was created by user '2022bcs0001-vishnu' with one commit. The README file contains a brief description of the lab assignment. The repository has 0 stars, 0 forks, and no releases or packages published.

Code | **Issues** | **Pull requests** | **Actions** | **Projects** | **Wiki** | **Security** | **Insights** | **Settings**

LAB_2 Public

main · 1 Branch · 0 Tags

Go to file Add file Code

2022bcs0001-vishnu Initial commit f1691aa · now 1 Commit

README.md Initial commit now

README

LAB_2

We will be doing our lab 2 for the course CSS 426. The visibility of this repository is going to be kept public.. I am going to be adding a Readme as well so as to help understand how this works.

About

We will be doing our lab 2 for the course CSS 426. The visibility of this repository is going to be kept public.. I am going to be adding a Readme as well so as to help understand how this works.

Readme

Activity

0 stars

0 watching

0 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Cloned the repository and adding the requirements file

The screenshot shows a Windows terminal window titled 'LAB_2' with a dark theme. It displays the command 'git clone https://github.com/2022bcs0001-vishnu/LAB_2.git' and its execution output, which shows the cloning process completed successfully. The terminal also shows the user navigating into the cloned repository directory with the command 'cd LAB_2'.

```
git clone https://github.com/2022bcs0001-vishnu/LAB_2.git
Cloning into 'LAB_2'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
> cd LAB_2
```

pwsh MEM: 63% | 9/15GB 31ms

10:29 | C: → main

System tray icons: Search, Task View, File Explorer, Camera, Spotify, WhatsApp, Task Manager, Power, Network, Battery, Date/Time.

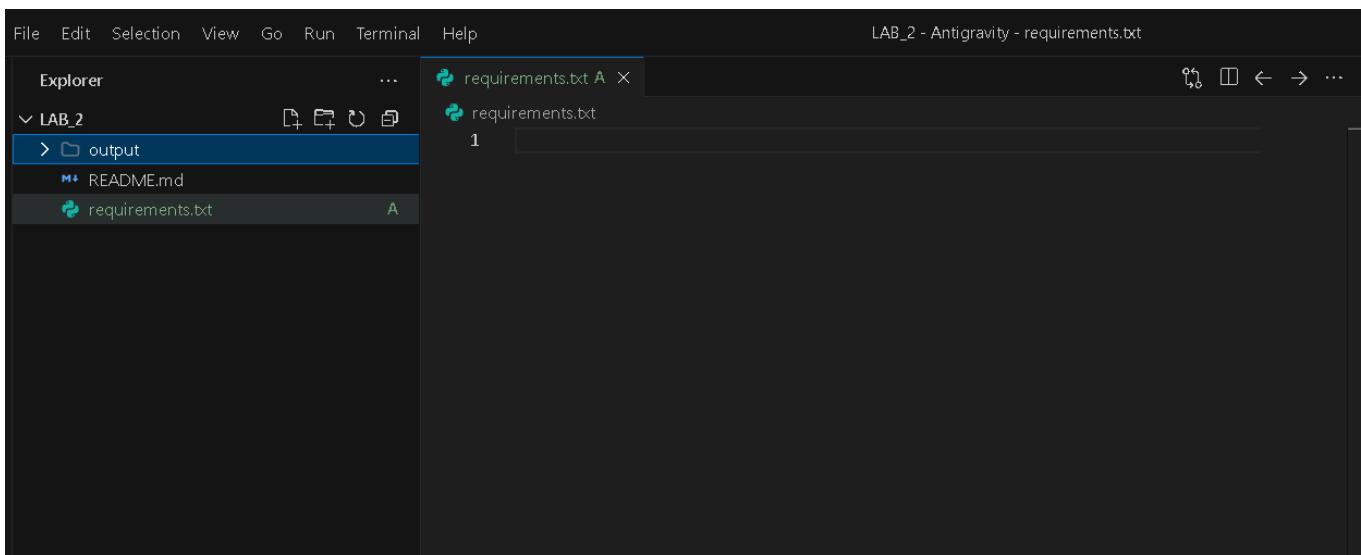
```
> git clone https://github.com/2022bcs0001-vishnu/LAB_2.git
Cloning into 'LAB_2' ...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
> cd LAB_2
> git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    requirements.txt

nothing added to commit but untracked files present (use "git add" to track)
> git add .
> git status
On branch main
Your branch is up to date with 'origin/main'.

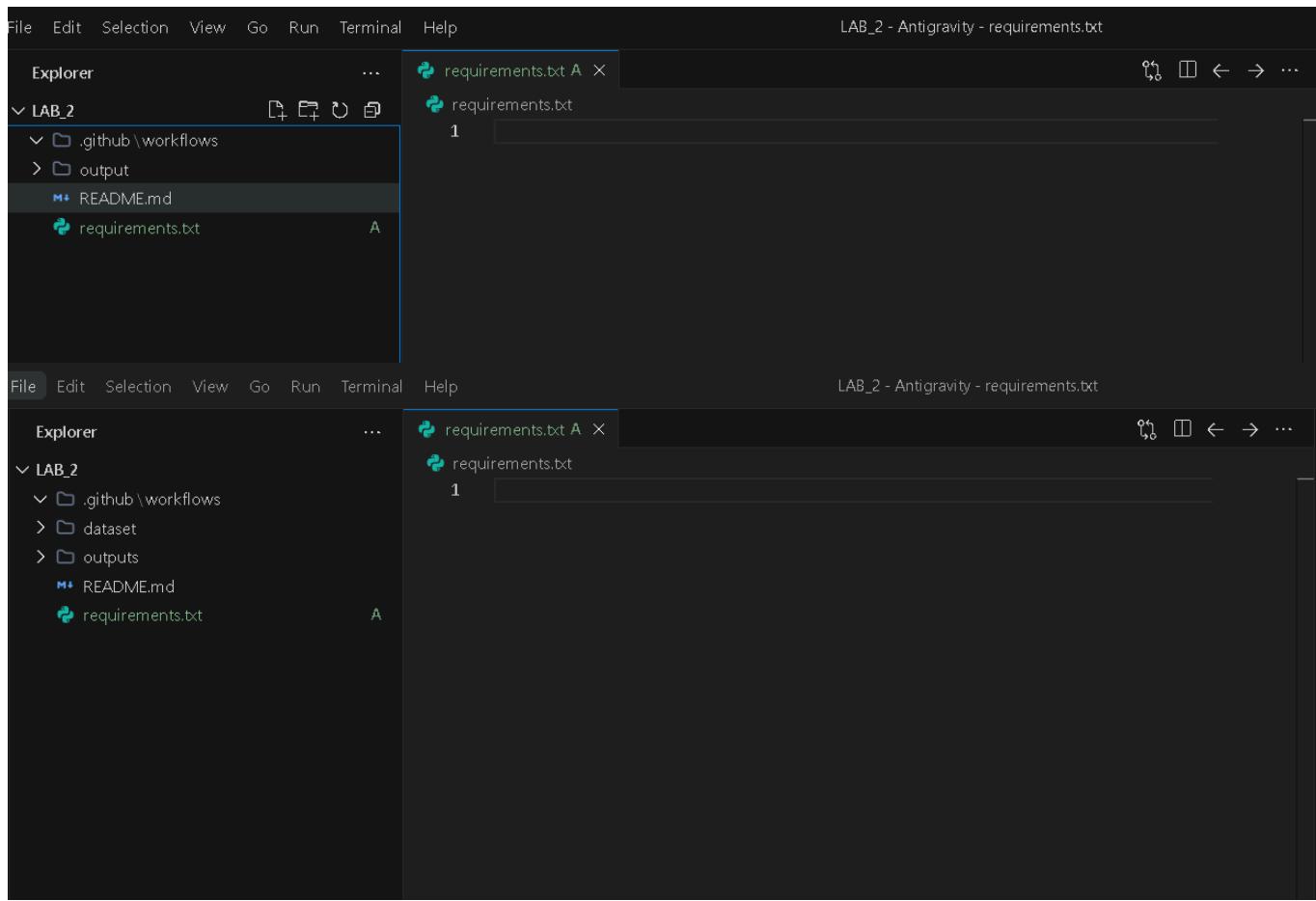
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   requirements.txt
```

pwsh MEM: 82% | 12/15GB 105ms
10:35 | C: → 📂 → 📂 → 📂 → 📂 → 📂 → 📂 → LAB_2



```
> mkdir -p .github/workflows

Directory: C:\Obsidians\Academics\2026\Subs\CSS426\LABS\LAB_2\.github
```



Task 2: Training Script

The training script must:

1. Load the dataset
2. Apply pre-processing and feature selection
3. Train the selected model
4. Evaluate the model using:
 - MSE
 - R² Score
5. Save:
 - The trained model to a file
 - Evaluation metrics to a JSON file
6. Print metrics to standard output

Output

Imports

```
[1] ✓ 3s
import pandas as pd
import json
import joblib
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error,r2_score
```

Dataset

```
[2] ✓ 0s
url="https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv"
df=pd.read_csv(url,sep=";")
```

Pre-processing and Feature Selection

```
[3] ✓ 0s
x=df.drop("quality",axis=1)
y=df["quality"]

fs=["alcohol","volatile acidity","sulphates","density"]
x=x[fs]

xtr,xts,ytr,yts=train_test_split(x,y,test_size=0.2,random_state=42)

sc=StandardScaler()
xtr=sc.fit_transform(xtr)
xts=sc.transform(xts)
```

Training

```
[4] ✓ 1s
m=RandomForestRegressor(n_estimators=200,random_state=42)
m.fit(xtr,ytr)
```

RandomForestRegressor

```
RandomForestRegressor(n_estimators=200, random_state=42)
```

Evaluate

```
[5] ✓ 0s
yp=m.predict(xts)

mse=mean_squared_error(yts,yp)
r2=r2_score(yts,yp)

print("mse:",mse)
print("r2:",r2)

mse: 0.36756039062500007
r2: 0.4375560901985983
```

Saving the outputs

```
[6] ✓ 0s
joblib.dump(m,"model.joblib")

res={
    "model":"RandomForestRegressor",
    "features":fs,
    "n_estimators":200,
    "mse":mse,
    "r2":r2
}

with open("results.json","w") as f:
    json.dump(res,f,indent=2)
```

```

1  {
2    "model": "RandomForestRegressor",
3    "features": [
4      "alcohol",
5      "volatile acidity",
6      "sulphates",
7      "density"
8    ],
9    "n_estimators": 200,
10   "mse": 0.36756039062500007,
11   "r2": 0.4375560901985983
12 }

```

```

> git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
(use "git restore --staged <file>..." to unstage)
  new file: requirements.txt

Untracked files:
(use "git add <file>..." to include in what will be committed)
  2022BCS0001_CSS_426_L2.ipynb
  dataset/
  outputs/

> git add .
warning: in the working copy of '2022BCS0001_CSS_426_L2.ipynb', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'dataset/winequality-red.csv', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'outputs/results/results.json', LF will be replaced by CRLF the next time Git touches it
> git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
(use "git restore --staged <file>..." to unstage)
  new file: 2022BCS0001_CSS_426_L2.ipynb
  new file: dataset/winequality-red.csv
  new file: outputs/models/model.joblib
  new file: outputs/results/results.json
  new file: requirements.txt

```

Staged the files

Task 3: GitHub Actions Workflow

You must create a GitHub Actions workflow that:

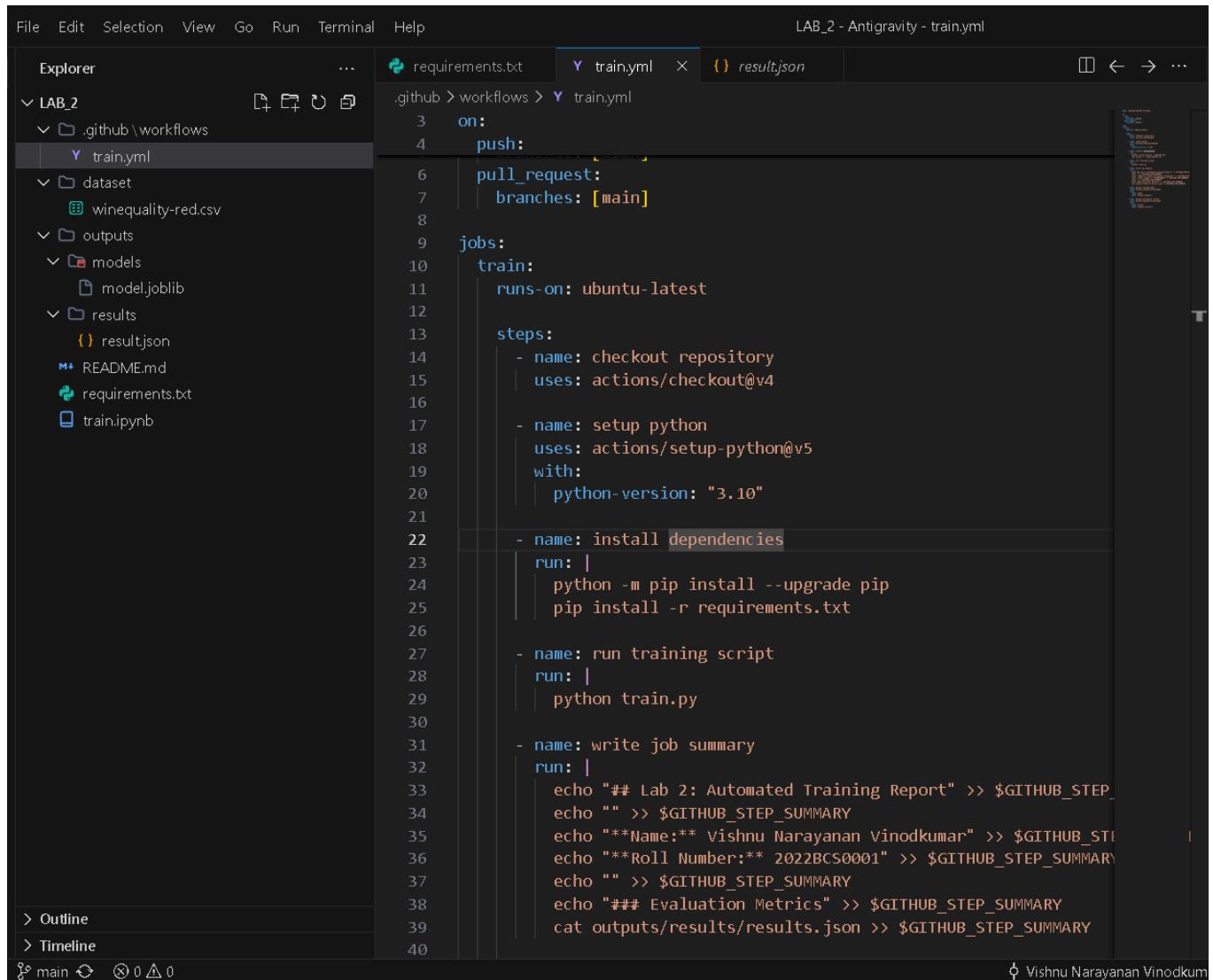
1. Triggers on:
 - Push to the main branch
 - Pull request on main branch
2. Sets up:
 - Python environment
 - Required dependencies
3. Executes the training script
4. Captures output metrics

5. Writes evaluation results to the Job Summary (The summary should also contain your name and roll number)

6. Uploads artifacts

Note: Hyperparameters, pre-processing steps, and feature selection MUST NOT be controlled from the workflow file.

Output



```
LAB_2 - Antigravity - train.yml
File Edit Selection View Go Run Terminal Help
Explorer ... requirements.txt train.yml result.json
.github > workflows > train.yml
3   on:
4     push:
5       pull_request:
6         branches: [main]
7
8   jobs:
9     train:
10       runs-on: ubuntu-latest
11
12       steps:
13         - name: checkout repository
14           uses: actions/checkout@v4
15
16         - name: setup python
17           uses: actions/setup-python@v5
18           with:
19             python-version: "3.10"
20
21         - name: install dependencies
22           run:
23             | python -m pip install --upgrade pip
24             | pip install -r requirements.txt
25
26         - name: run training script
27           run:
28             | python train.py
29
30         - name: write job summary
31           run:
32             | echo "## Lab 2: Automated Training Report" >> $GITHUB_STEP_SUMMARY
33             | echo "" >> $GITHUB_STEP_SUMMARY
34             | echo "**Name:** Vishnu Narayanan Vinodkumar" >> $GITHUB_STEP_SUMMARY
35             | echo "**Roll Number:** 2022BCS0001" >> $GITHUB_STEP_SUMMARY
36             | echo "" >> $GITHUB_STEP_SUMMARY
37             | echo "## Evaluation Metrics" >> $GITHUB_STEP_SUMMARY
38             | cat outputs/results/results.json >> $GITHUB_STEP_SUMMARY
39
40
```

Vishnu Narayanan Vinodkum

added the train.yml

This is the first commit of LAB_2 repository in the main branch. I ha... 42852ee · 16 minutes ago 2 Commits

dataset This is the first commit of LAB_2 repository in the main bra... 16 minutes ago

outputs This is the first commit of LAB_2 repository in the main bra... 16 minutes ago

README.md Initial commit 45 minutes ago

requirements.txt This is the first commit of LAB_2 repository in the main bra... 16 minutes ago

train.ipynb This is the first commit of LAB_2 repository in the main bra... 16 minutes ago

LAB_2

We will be doing our lab 2 for the course CSS 426. The visibility of this repository is going to be kept public.. I am going to be adding a Readme as well so as to help understand how this works.

https://github.com/Vishnu0107/LAB_2

Pushed to the main branch

Triggered via push 1 minute ago

Vishnu0107 pushed → 6ae7c5b main

Status Success Total duration 29s Artifacts 2

train.yml

on: push

train 26s

train summary

...

Lab 2: Automated Training Report

Name: Vishnu Narayanan Vinodkumar

Roll Number: 2022BCS0001

Evaluation Metrics

```
{  
  "model": "RandomForestRegressor",  
  "features": [  
    "alcohol",  
    "volatile acidity",  
    "sulphates",  
    "density"  
  ],  
  "n_estimators": 200,  
  "mse": 0.36756039062500007,  
  "r2": 0.4375560901985983  
}
```

Job summary generated at run-time

train.yml

```
name: lab2-automated-training  
  
on:  
  push:  
    branches: [main]  
  pull_request:  
    branches: [main]  
  
jobs:  
  train:  
    runs-on: ubuntu-latest  
  
    steps:  
      - name: checkout repository  
        uses: actions/checkout@v4  
  
      - name: setup python  
        uses: actions/setup-python@v5  
        with:  
          python-version: "3.10"  
  
      - name: install dependencies  
        run: |
```

```

python -m pip install --upgrade pip
pip install -r requirements.txt

- name: run training script
  run: |
    python train.py

- name: write job summary
  run: |
    echo "## Lab 2: Automated Training Report" >> $GITHUB_STEP_SUMMARY
    echo "" >> $GITHUB_STEP_SUMMARY
    echo "**Name:** Vishnu Narayanan Vinodkumar" >> $GITHUB_STEP_SUMMARY
    echo "**Roll Number:** 2022BCS0001" >> $GITHUB_STEP_SUMMARY
    echo "" >> $GITHUB_STEP_SUMMARY
    echo "### Evaluation Metrics" >> $GITHUB_STEP_SUMMARY
    cat outputs/results/result.json >> $GITHUB_STEP_SUMMARY

- name: upload trained model
  uses: actions/upload-artifact@v4
  with:
    name: model
    path: outputs/model/

- name: upload evaluation results
  uses: actions/upload-artifact@v4
  with:
    name: result
    path: outputs/results/

```

Task 4: Artifact Storage

Each workflow run must upload artifacts, including:

Required Artifacts

1. Trained model file
2. Results JSON file containing:
 - MSE
 - R² score

Artifacts must be downloadable from the GitHub Actions run page.

Output

Artifacts			
Produced during runtime			
Name	Size	Digest	
📦 model	1.58 MB	sha256:3ade5913c7b3143b7a49f14b95ddaf629a3...	🔗 ⏪ 🗑
📦 result	291 Bytes	sha256:a870aa5c2c2669aa6e321272bc9064f5ec4...	🔗 ⏪ 🗑

Task 5: Run Multiple Experiments

You must perform all experiments as done in Lab 1 , where each experiment involves:

- Editing the training script (e.g., model type, hyperparameters, preprocessing)
- Committing the changes with a meaningful commit message (Eg. “Model- Lasso, alpha-0.1, testsplit-0.2, pre-processing – scalar”)
- Pushing to GitHub to trigger a workflow run

Each run represents one experiment.

Output

The screenshot shows a GitHub Actions workflow run summary for a repository named 'lab2-automated-training'. The workflow has triggered via a push from user 'Vishnu0107' on branch 'main'. The run was successful, taking 29 seconds. It produced 2 artifacts. The workflow file is named 'train.yml' and contains a single job named 'train' which completed successfully in 24 seconds. The interface includes standard GitHub navigation and search bars at the top, and a sidebar on the left showing other workflow runs and repository details.

Summary

All jobs

train

Run details

Usage

Workflow file

```
"split": "80/20",
"mse": 0.3682457611173189,
"r2": 0.43650733067718506
},
{
"id": "EXP-05",
"model": "Lasso",
"params": "alpha=0.1",
"preprocessing": "standardscaler",
"features": "all",
"split": "80/20",
"mse": 0.4392249149065927,
"r2": 0.327894450209431
}
]
```

Job summary generated at run-time

Artifacts

Produced during runtime

Name	Size	Digest	Download	Remove
model	664 Bytes	sha256:28d395f1e1979507cf5ebb4500178a50ec...	Download	Remove
result	477 Bytes	sha256:58638f209698bac089c546a58a72ba78bf...	Download	Remove

train summary ...

Lab 2: Automated Training Report

Name: Vishnu Narayanan Vinodkumar
Roll Number: 2022BCS0001

Evaluation Metrics

```
[{"id": "EXP-01", "model": "LinearRegression", "params": "default", "preprocessing": "none", "features": "all", "split": "80/20", "mse": 0.3900251439639545, "r2": 0.40318034127962254}, {"id": "EXP-02", "model": "Ridge", "params": "alpha=1.0"}]
```

The screenshot shows the GitHub Actions interface for a repository named 'lab2-automated-training'. On the left, there's a sidebar with 'Actions' selected, showing a list of workflows: 'All workflows' (selected), 'lab2-automated-training', 'Management' (with sub-options: Caches, Attestations, Runners, Usage metrics, Performance metrics), and 'New workflow'. The main area displays 'All workflows' with a search bar 'Filter workflow runs'. Below it, a table lists '8 workflow runs' with columns for 'Event', 'Status', 'Branch', and 'Actor'. The runs are as follows:

Event	Status	Branch	Actor
Updated the train.py	Success	main	Vishnu0107
Updated the train.py	Success	main	Vishnu0107
Running lab 1 experiments and adding train_2.py	Success	main	Vishnu0107
Trigger CI	Success	main	Vishnu0107
Updated folder name	Success	main	Vishnu0107
Updated folder name	Success	main	Vishnu0107

Task 6: Analysis

Students must answer:

1. How did GitHub Actions improve experiment reproducibility?
2. How easy was it to compare results across runs?
3. What role does Git commit history play in experiment tracking?
4. What were the benefits of this approach compared to Lab 1.
5. What limitations does this approach have?

Output

1. How did GitHub Actions improve experiment reproducibility?

GitHub Actions ensured that every experiment was executed in a clean and consistent environment. Each run followed the same steps for installing dependencies, training the model, and reporting metrics, which eliminated differences caused by local machine settings.

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

Comparing results was easier than in Lab 1 because each experiment produced standardized outputs and metrics. The GitHub Actions job summary and saved artifacts allowed results from different runs to be reviewed and compared systematically.

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

Git commit history acted as a record of each experiment. Meaningful commit messages clearly documented changes in models, hyperparameters, preprocessing steps, and data splits, making it possible to trace results back to the exact code used.

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

Unlike manual tracking in Lab 1, this approach reduced human error and improved traceability. Experiments were automatically executed, metrics were consistently reported, and models and results were stored as artifacts for future reference.

5. What limitations does this approach have?

Experiments still required manual code edits for each run, which can become inefficient as the number of experiments grows. Additionally, GitHub Actions provides limited visualization and analysis tools compared to dedicated experiment tracking platforms.