

CSS 426 - MLOps

Name : - VISHNU NARAYANAN VINODKUMAR

Roll Number :- 2022BCS0001

Date :- 09-01-2026

Batch :- 1

Lab :- 2

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

Type **Z** to search

OverviewRepositoriesProjectsPackagesStars



2022bcs0001-vishnu

2022bcs0001-vishnu · he/him

IIIT Kottayam - Lab account Vishnu Narayanan Vinodkumar

Edit profile

IIIT Kottayam

Joined 12 minutes ago

Popular repositories

You don't have any public repositories yet.

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

Settings

Type **Z** to search

2022bcs0001-vishnu (2022bcs0001-vishnu)

Your personal account

Go to your personal profile

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

Public profile

Name

2022bcs0001-vishnu

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

Public email

vishnu22bcs1@iiitkottayam.ac.in

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

Bio

IIIT Kottayam - Lab account
Vishnu Narayanan Vinodkumar

You can @mention other users and organizations to link to them.


Pronouns

he/him

URL

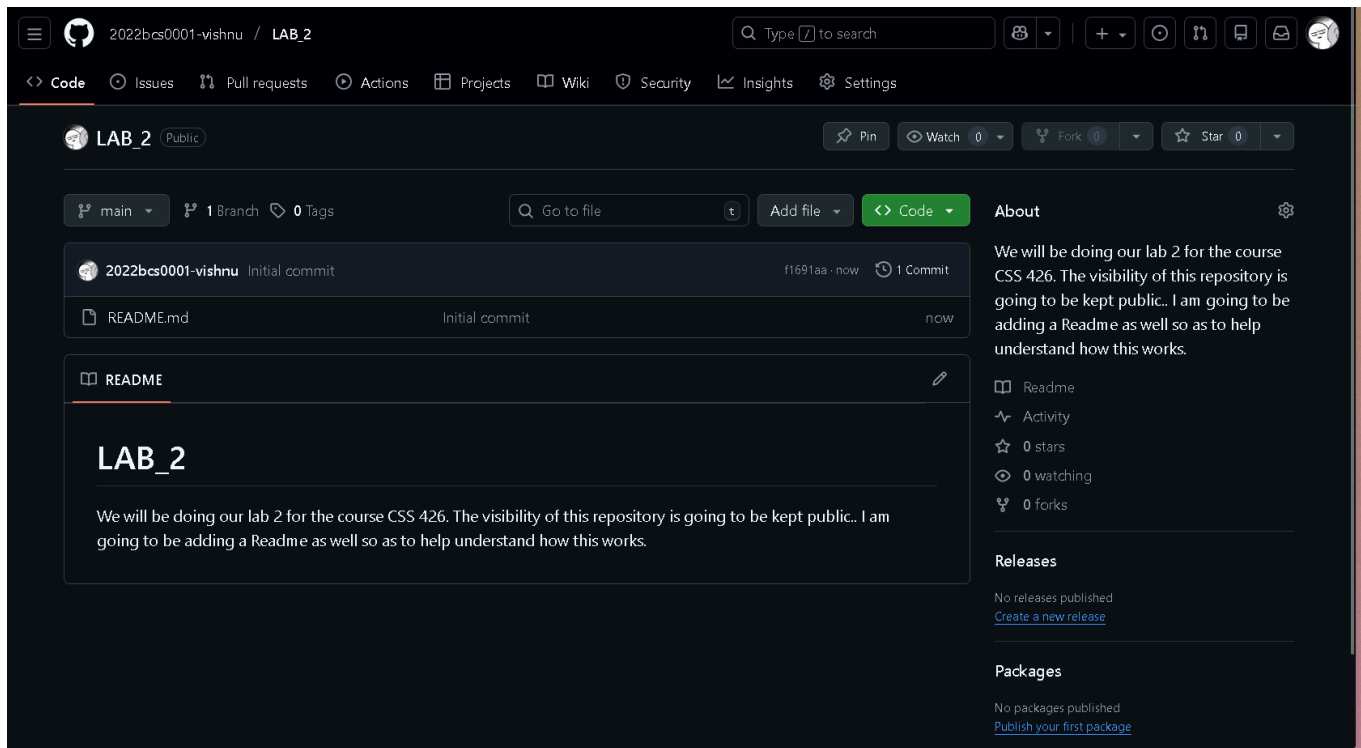
Social accounts

Profile picture

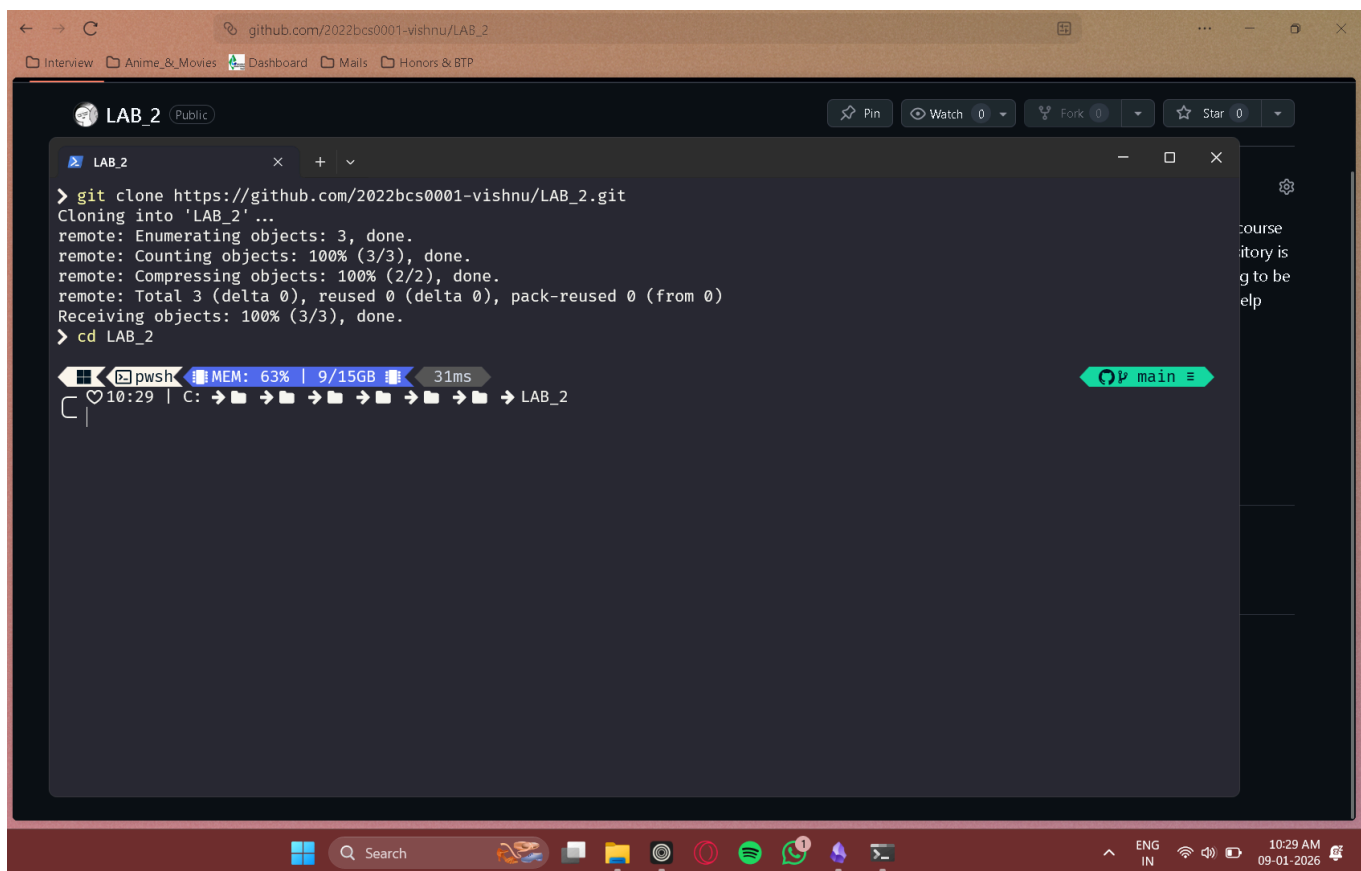


Edit

Repository named LAB_2 has been made



Cloned the repository and adding the requirements file



```
2022bcs0001-vishnu / LAB_2
Type to search

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
> 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

10:35 | C: \> LAB_2
```

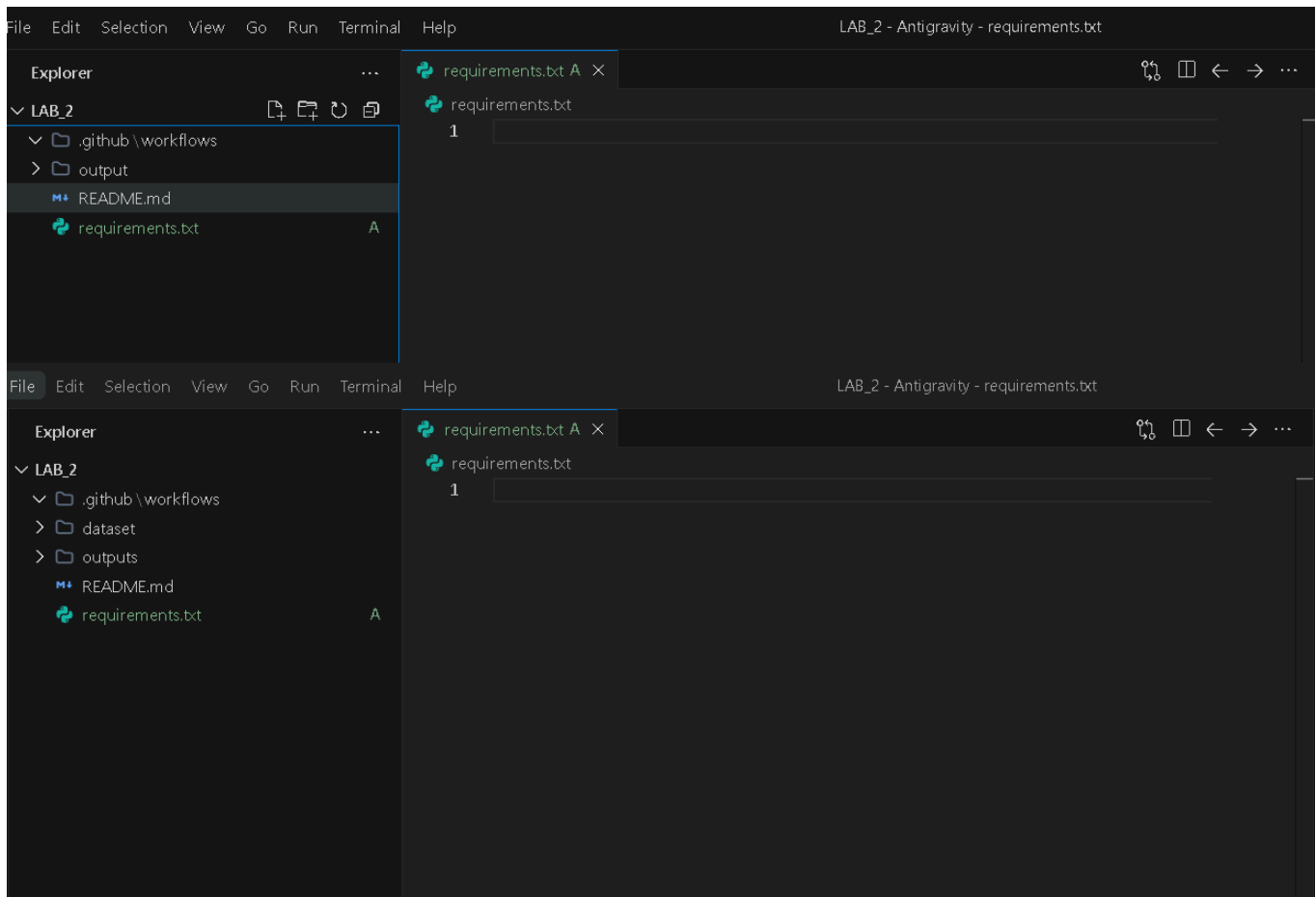
```
File Edit Selection View Go Run Terminal Help
LAB_2 - Antigravity - requirements.txt

Explorer
LAB_2
  > output
  README.md
  requirements.txt

requirements.txt
1
```

```
> 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
 - R^2 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)
```

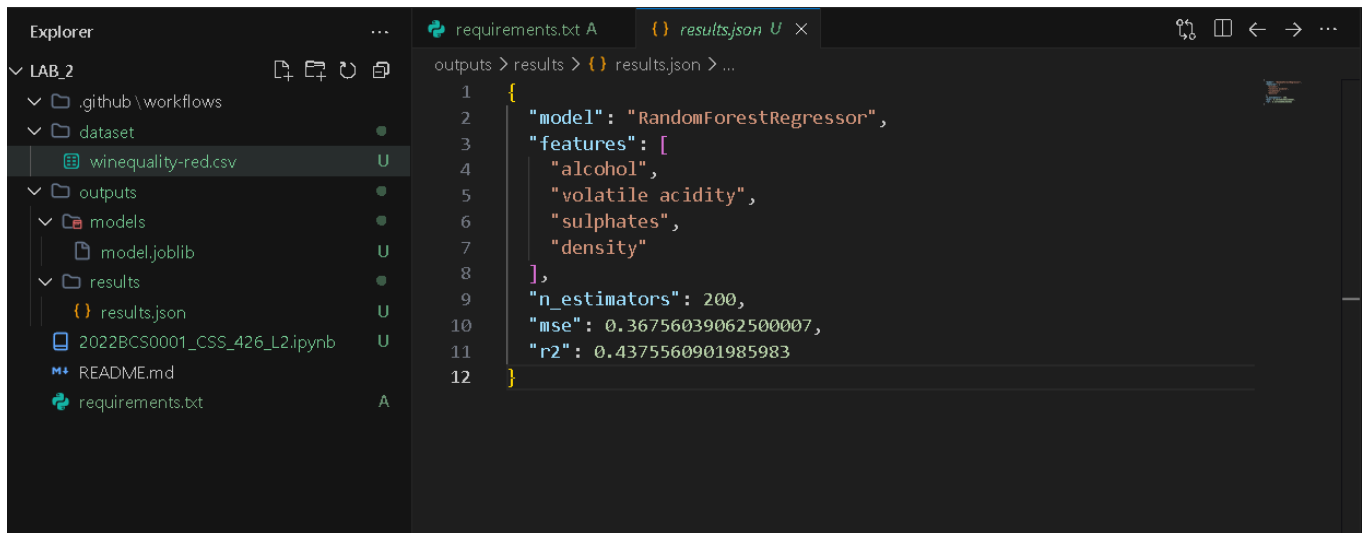
```
mse: 0.36756039062500007
r2: 0.4375560901985983
```

Saving the outputs

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

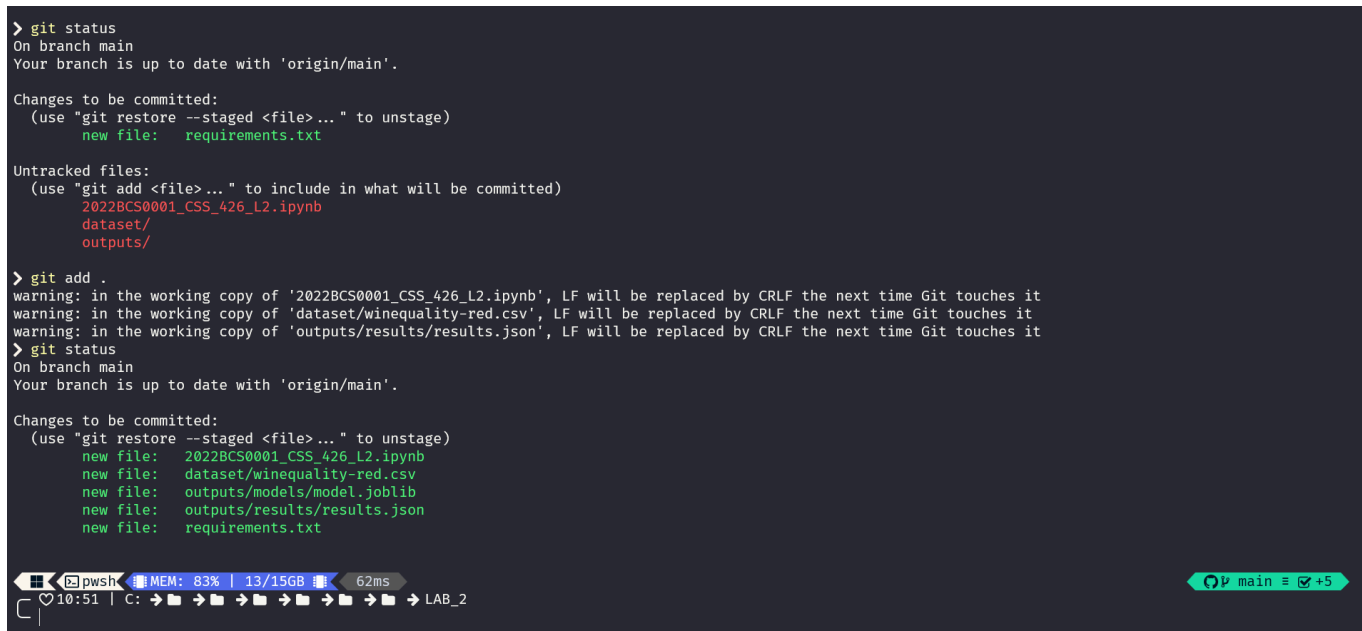
res={
    "model":"RandomForestRegressor",
    "features":fs,
    "n_estimators":200,
    "mse":mse,
    "r2":r2
}

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



The screenshot shows the VS Code interface. The Explorer on the left shows a project structure with folders like .github, dataset, outputs, models, and results. The file results.json is selected. The Output window on the right shows the content of results.json, which is a JSON object containing model details.

```
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 }
```



The screenshot shows a terminal window with the following output:

```
> 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

- Writes evaluation results to the Job Summary (The summary should also contain your name and roll number)
- Uploads artifacts

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

Output

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

added the train.yml

2022bcs0001-vishnu / LAB_2

Type [7] to search

+ -

+

+

+

+

+

+

+

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

LAB_2

Public

Watch 0

Fork 0

Star 0

main

1 Branch

0 Tags

Go to file

t

Add file

<> Code

About

Vishnu0107

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

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

Report repository

Releases

No releases published

Create a new release

Packages

No packages published

Pushed to the main branch

2022bcs0001-vishnu / LAB_2

Type [7] to search

+ -

+

+

+

+

+

+

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

lab2-automated-training

Trigger CI #5

Re-run all jobs

...

Summary

All jobs

train

Run details

Usage

Workflow file

Triggered via push 1 minute ago

Status

Total duration

Artifacts

Vishnu0107 pushed · 6ae7c5b · main

Success

29s

2

train.yml

on: push

train

26s

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
 - R^2 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...	  

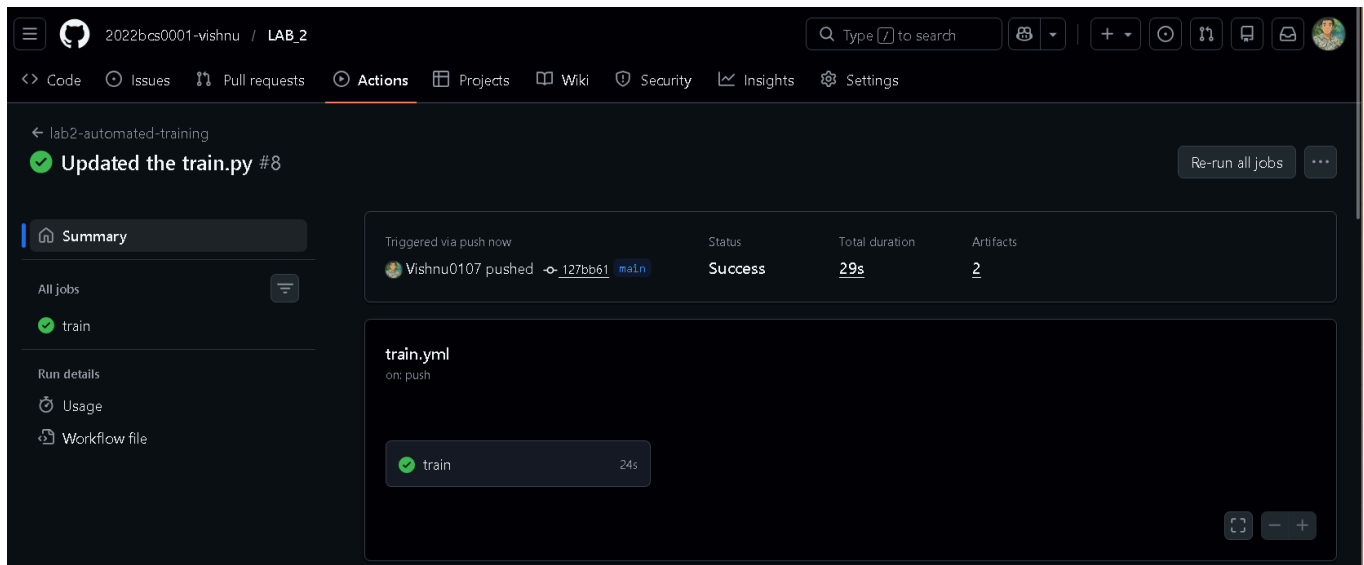
Task 5: Run Multiple Experiments

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 the GitHub Actions interface for a workflow named 'lab2-automated-training'. The workflow is triggered by a push to the 'main' branch. The status is 'Success' with a total duration of 29s and 2 artifacts. The 'train' job is shown as completed with a duration of 24s.

Summary:

- Workflow: lab2-automated-training
- Commit: Vishnu0107 pushed -> 127bb61 main
- Status: Success
- Total duration: 29s
- Artifacts: 2
- Jobs: train (24s)

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			
model	664 Bytes	sha256:28d395f1e1979507cfc5ebb4509178a50ec...			
result	477 Bytes	sha256:58638f209698bac089c546a58a72ba78bf0...			

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 displays the GitHub Actions interface for a repository named '2022bcs0001-vishnu / LAB_2'. The left sidebar shows the 'Actions' tab with a 'New workflow' button and a list of workflow runs for 'lab2-automated-training'. The main area shows a table of 8 workflow runs, each with a status icon (green checkmark for success, red X for failure), a title, a commit hash, a branch name, and a completion time. The runs are as follows:

Status	Event	Commit	Branch	Actor	Time
Success	Updated the train.py	lab2-automated-training #8: Commit 127bb61 pushed by Vishnu0107	main		2 minutes ago, 29s
Failure	Updated the train.py	lab2-automated-training #7: Commit 5594a95 pushed by Vishnu0107	main		6 minutes ago, 30s
Failure	Running lab 1 experiments and adding train_2.py	lab2-automated-training #6: Commit 97737bd pushed by Vishnu0107	main		8 minutes ago, 30s
Success	Trigger CI	lab2-automated-training #5: Commit 6ae7c5b pushed by Vishnu0107	main		19 minutes ago, 29s
Success	Updated folder name	lab2-automated-training #4: Commit 6ad34ed pushed by Vishnu0107	main		20 minutes ago, 29s
Failure	Updated folder name	lab2-automated-training #3: Commit 3cbd377 pushed by Vishnu0107	main		22 minutes ago, 27s

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.