Solution for M4 Mini Project-1

Bike Rental Prediction

Continuous Integration

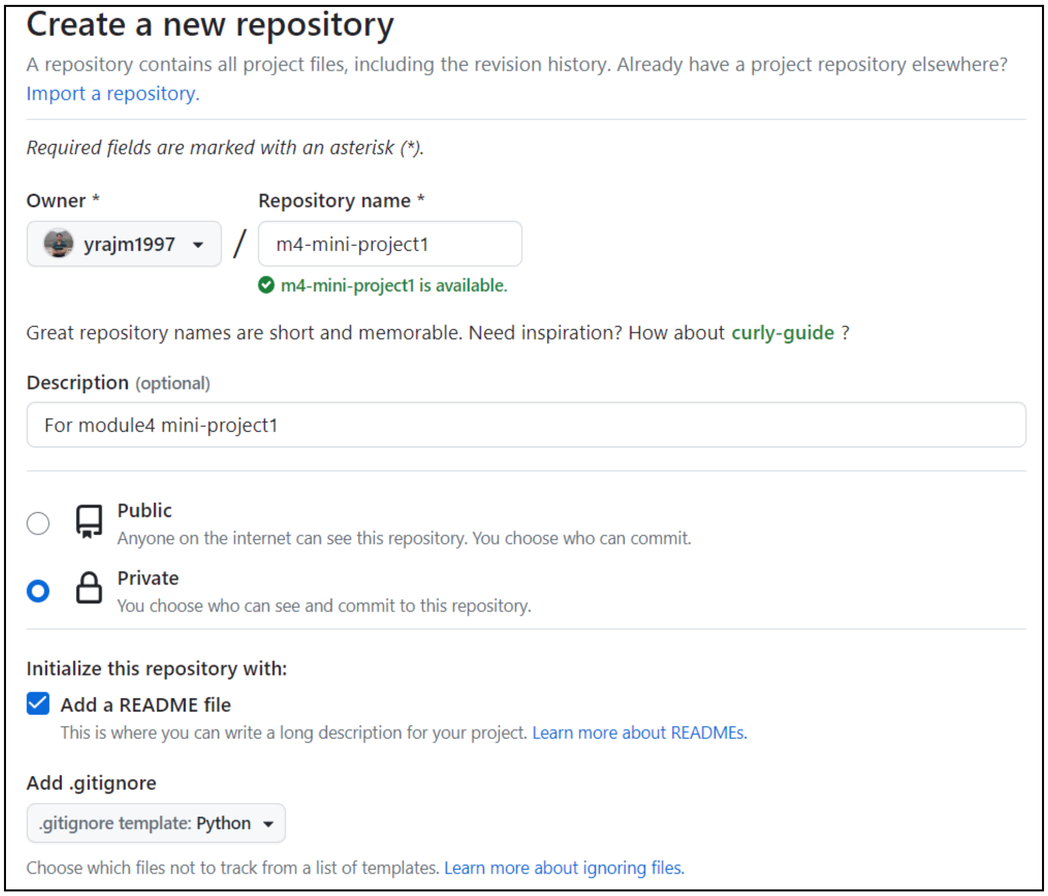
For this project, we will build a GitHub Actions workflow to automate model training, testing, linting, and formatting steps for the bike rental count prediction system. Please refer to Module 4 - AST 1 for this mini-project.

Step 1: Download project folder in your local system:

1.1 Download the given project folder '*bikeshare_project*' on to your system

Step 2: On your GitHub account, create a new repository: (1 point)

2.1 Create a new repository to store files related to this mini-project



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner * : yrajm1997 / Repository name * : m4-mini-project1

m4-mini-project1 is available.

Great repository names are short and memorable. Need inspiration? How about curly-guide ?

Description (optional): For module4 mini-project1

Public [✓ Private]

Public — Anyone on the internet can see this repository. You choose who can commit.

Private — You choose who can see and commit to this repository.

Initialize this repository with:

✓ Add a README file

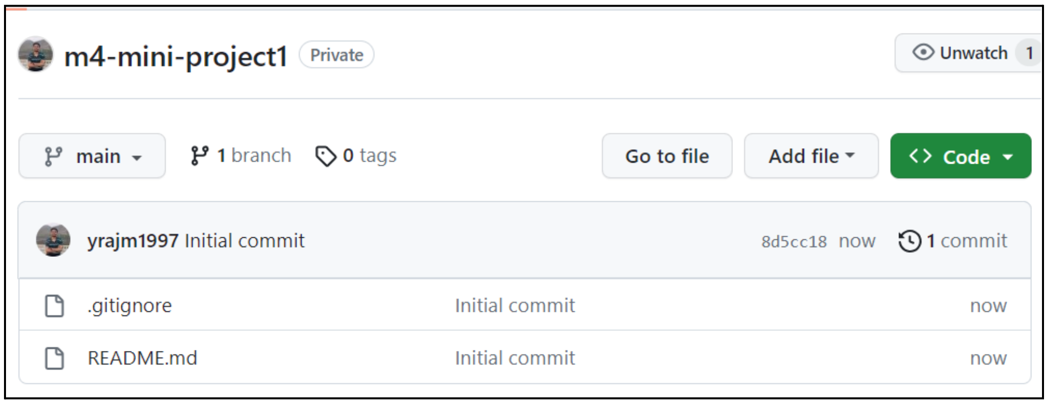
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: Python

Choose which files not to track from a list of templates. Learn more about ignoring files.

Created Repository:



yrajm1997 Initial commit 8d5cc18 now 1 commit

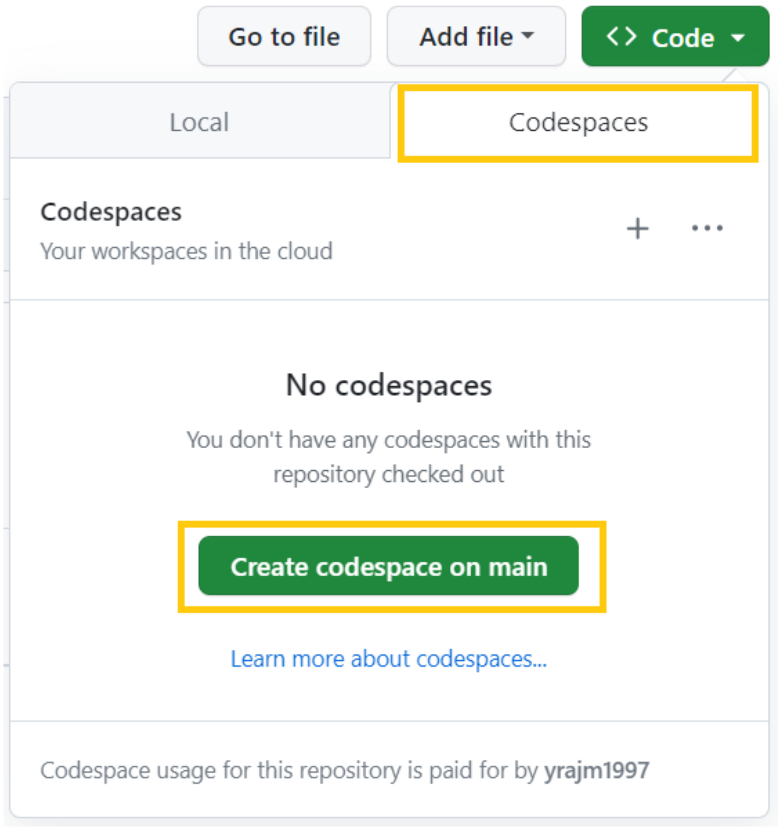
.gitignore Initial commit now

README.md Initial commit now

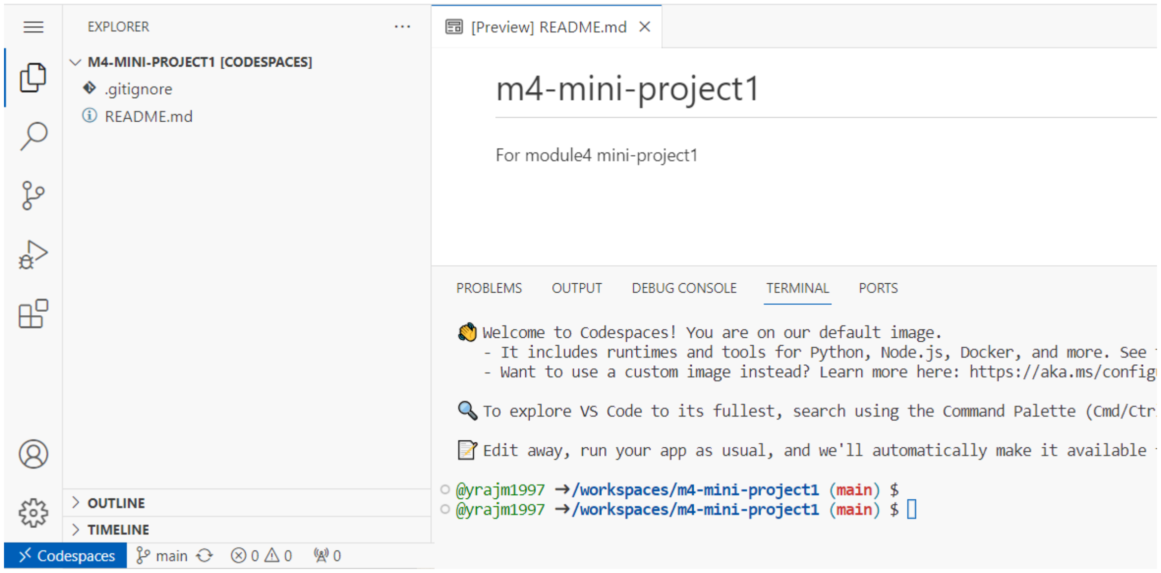
Step 3: Setup a Cloud development environment: (2 points)

3.1 Setup a cloud development environment, such as AWS Cloud9, or GitHub Codespaces.

1. Select the Code > *Codespaces* tab, and select *Create codespace on main*.


Go to file
Add file
Code
Local
Codespaces
Codespaces
Your workspaces in the cloud
+
...
No codespaces
You don't have any codespaces with this repository checked out
Create codespace on main
Learn more about codespaces...
Codespace usage for this repository is paid for by yrajm1997

2. Codespace created.



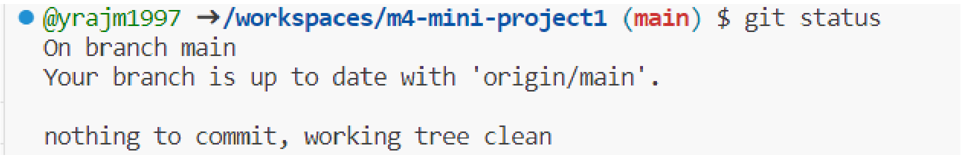
m4-mini-project1

For module4 mini-project1

```
👋 Welcome to Codespaces! You are on our default image.
   - It includes runtimes and tools for Python, Node.js, Docker, and more. See
   - Want to use a custom image instead? Learn more here: https://aka.ms/config

🔍 To explore VS Code to its fullest, search using the Command Palette (Cmd/Ctr

📝 Edit away, run your app as usual, and we'll automatically make it available

@yrajm1997 ➜/workspaces/m4-mini-project1 (main) $
@yrajm1997 ➜/workspaces/m4-mini-project1 (main) $
```

3.2 Authenticate the communication between Cloud dev environment to GitHub Repository by SSH method

1. Since we opened Codespace through the repository itself, the communication between cloud dev environment and GitHub repository is already there.
2. Check the status

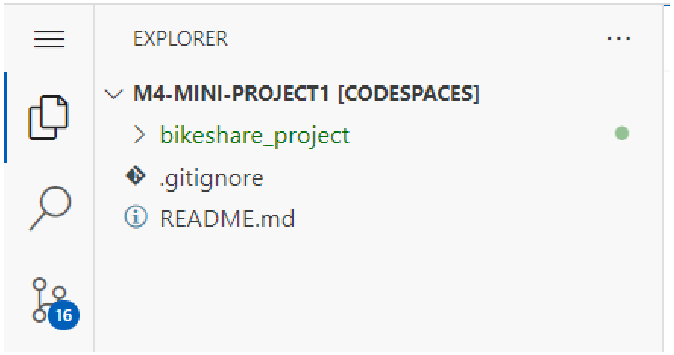


```
@yrajm1997 ➜/workspaces/m4-mini-project1 (main) $ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Step 4: Clone the remote repository in cloud dev environment: (1 point)

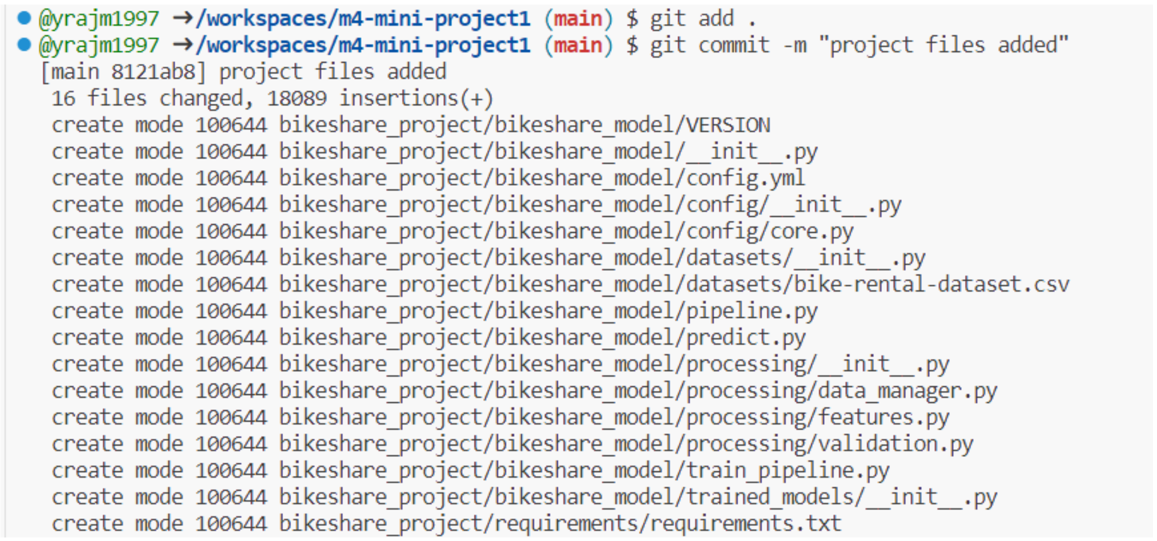
4.1 Clone the remote repository in your cloud environment (only for Cloud9).
4.2 Add the downloaded project folder to this cloned repository.

1. Drag your folder from your system and drop it into the Codespace Explorer pane. The folder will be added.



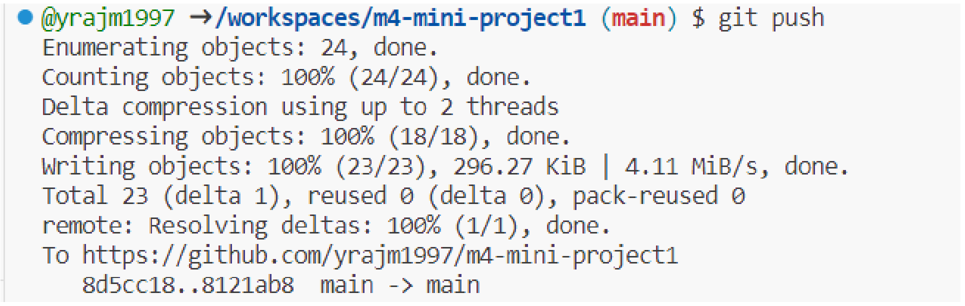
4.3 Finally, push the changes into the remote GitHub repository.

1. Perform git add and git commit



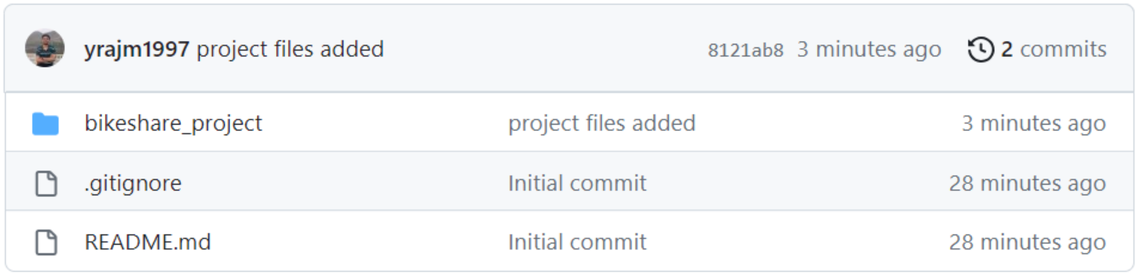
```
● @yrajm1997 ➜ /workspaces/m4-mini-project1 (main) $ git add .
● @yrajm1997 ➜ /workspaces/m4-mini-project1 (main) $ git commit -m "project files added"
  [main 8121ab8] project files added
   16 files changed, 18089 insertions(+)
   create mode 100644 bikeshare_project/bikeshare_model/VERSION
   create mode 100644 bikeshare_project/bikeshare_model/__init__.py
   create mode 100644 bikeshare_project/bikeshare_model/config.yml
   create mode 100644 bikeshare_project/bikeshare_model/config/__init__.py
   create mode 100644 bikeshare_project/bikeshare_model/config/core.py
   create mode 100644 bikeshare_project/bikeshare_model/datasets/__init__.py
   create mode 100644 bikeshare_project/bikeshare_model/datasets/bike-rental-dataset.csv
   create mode 100644 bikeshare_project/bikeshare_model/pipeline.py
   create mode 100644 bikeshare_project/bikeshare_model/predict.py
   create mode 100644 bikeshare_project/bikeshare_model/processing/__init__.py
   create mode 100644 bikeshare_project/bikeshare_model/processing/data_manager.py
   create mode 100644 bikeshare_project/bikeshare_model/processing/features.py
   create mode 100644 bikeshare_project/bikeshare_model/processing/validation.py
   create mode 100644 bikeshare_project/bikeshare_model/train_pipeline.py
   create mode 100644 bikeshare_project/bikeshare_model/trained_models/__init__.py
   create mode 100644 bikeshare_project/requirements/requirements.txt
```

2. Then git push



```
● @yrajm1997 ➜ /workspaces/m4-mini-project1 (main) $ git push
  Enumerating objects: 24, done.
  Counting objects: 100% (24/24), done.
  Delta compression using up to 2 threads
  Compressing objects: 100% (18/18), done.
  Writing objects: 100% (23/23), 296.27 KiB | 4.11 MiB/s, done.
  Total 23 (delta 1), reused 0 (delta 0), pack-reused 0
  remote: Resolving deltas: 100% (1/1), done.
  To https://github.com/yrajm1997/m4-mini-project1
     8d5cc18..8121ab8  main -> main
```

3. Same should reflect in the UI.



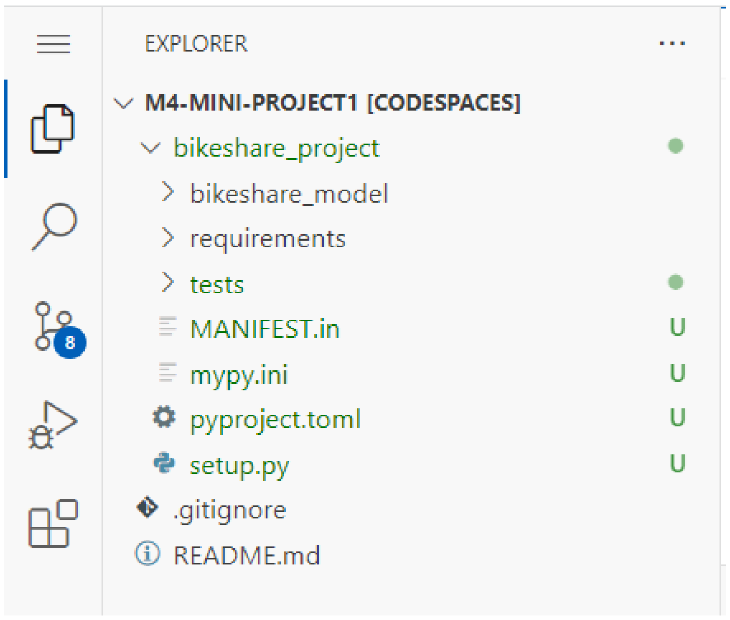
yrajm1997 project files added	8121ab8 3 minutes ago	**2** commits
bikeshare_project	project files added	3 minutes ago
.gitignore	Initial commit	28 minutes ago
README.md	Initial commit	28 minutes ago

Step 5: Add files related to testing, linting, and code formatting tools: (1 point)

5.1 Add additional files related to testing, linting, and code formatting, such as *conftest.py*, *test_features.py*, *test_prediction.py*, *pyproject.toml*, *setup.py*, *MANIFEST.in*, and *mypy.ini*.

5.2 Add the additional necessary dependencies/libraries to requirements directory

1. Add additional testing files from the previous module's mini-projects. Drag and drop them in the codespace.

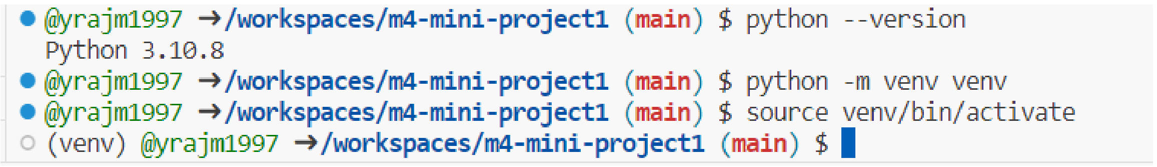


2. Similarly, add test_requirements.txt

Step 6: Run your model training, testing, linting, and formatting steps on the Cloud environment: (2 points)

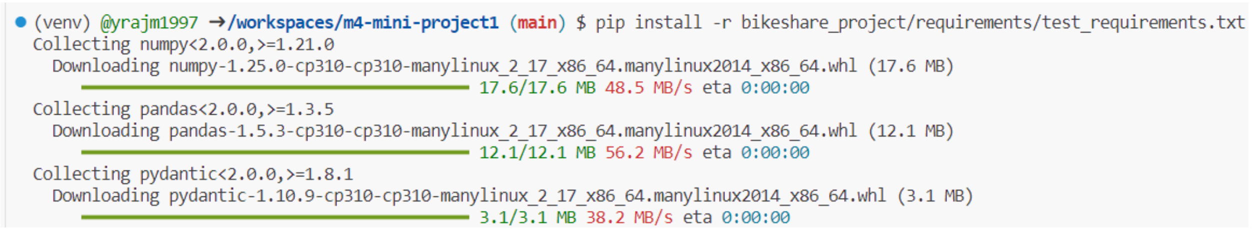
6.1 In the cloud environment, create a virtual environment, and execute commands to train the model, run test cases, linting, and formatting.
6.2 If the errors persist, debug your code and re-run.

1. Create a virtual environment and activate it.



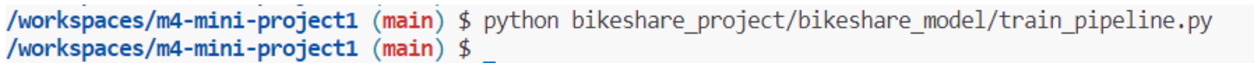
```
● @yrajm1997 →/workspaces/m4-mini-project1 (main) $ python --version
  Python 3.10.8
● @yrajm1997 →/workspaces/m4-mini-project1 (main) $ python -m venv venv
● @yrajm1997 →/workspaces/m4-mini-project1 (main) $ source venv/bin/activate
○ (venv) @yrajm1997 →/workspaces/m4-mini-project1 (main) $ 
```

2. Install requirements



```
● (venv) @yrajm1997 →/workspaces/m4-mini-project1 (main) $ pip install -r bikeshare_project/requirements/test_requirements.txt
  Collecting numpy<2.0.0,>=1.21.0
    Downloading numpy-1.25.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (17.6 MB)
       ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 17.6/17.6 MB 48.5 MB/s eta 0:00:00
  Collecting pandas<2.0.0,>=1.3.5
    Downloading pandas-1.5.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (12.1 MB)
       ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.1/12.1 MB 56.2 MB/s eta 0:00:00
  Collecting pydantic<2.0.0,>=1.8.1
    Downloading pydantic-1.10.9-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.1 MB)
       ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.1/3.1 MB 38.2 MB/s eta 0:00:00
```

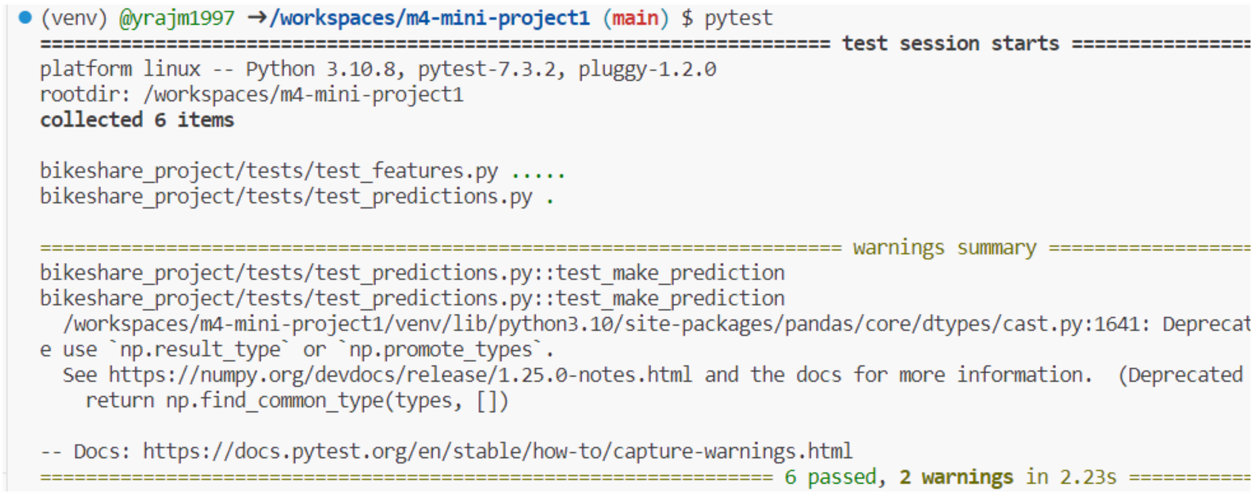
3. Train model



```
/workspaces/m4-mini-project1 (main) $ python bikeshare_project/bikeshare_model/train_pipeline.py
/workspaces/m4-mini-project1 (main) $ _
```

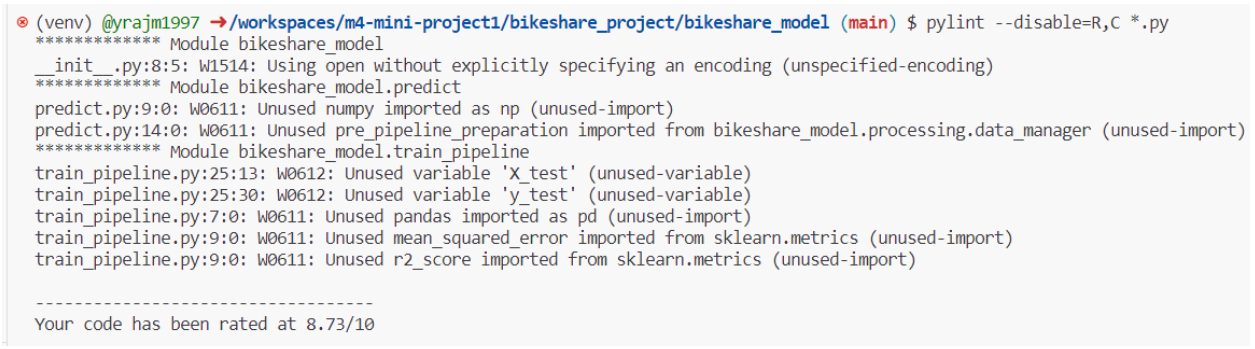
Trained model is stored in the folder now.

4. Run test cases



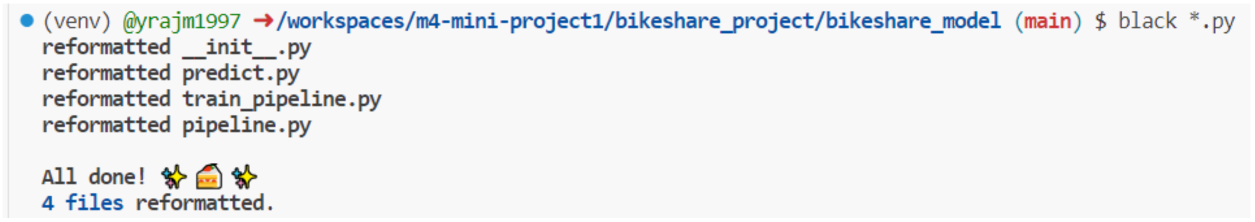
```
● (venv) @yrajm1997 →/workspaces/m4-mini-project1 (main) $ pytest
===================================================================== test session starts ================
platform linux -- Python 3.10.8, pytest-7.3.2, pluggy-1.2.0
rootdir: /workspaces/m4-mini-project1
collected 6 items

bikeshare_project/tests/test_features.py .....
bikeshare_project/tests/test_predictions.py .

====================================================================== warnings summary ==================
bikeshare_project/tests/test_predictions.py::test_make_prediction
bikeshare_project/tests/test_predictions.py::test_make_prediction
  /workspaces/m4-mini-project1/venv/lib/python3.10/site-packages/pandas/core/dtypes/cast.py:1641: Deprecat
e use `np.result_type` or `np.promote_types`.
  See https://numpy.org/devdocs/release/1.25.0-notes.html and the docs for more information.  (Deprecated
    return np.find_common_type(types, [])

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
=============================================================== 6 passed, 2 warnings in 2.23s ==========
```

5. Go to the bikeshare_model directory and run pylint command



```
⊗ (venv) @yrajm1997 →/workspaces/m4-mini-project1/bikeshare_project/bikeshare_model (main) $ pylint --disable=R,C *.py
************* Module bikeshare_model
__init__.py:8:5: W1514: Using open without explicitly specifying an encoding (unspecified-encoding)
************* Module bikeshare_model.predict
predict.py:9:0: W0611: Unused numpy imported as np (unused-import)
predict.py:14:0: W0611: Unused pre_pipeline_preparation imported from bikeshare_model.processing.data_manager (unused-import)
************* Module bikeshare_model.train_pipeline
train_pipeline.py:25:13: W0612: Unused variable 'X_test' (unused-variable)
train_pipeline.py:25:30: W0612: Unused variable 'y_test' (unused-variable)
train_pipeline.py:7:0: W0611: Unused pandas imported as pd (unused-import)
train_pipeline.py:9:0: W0611: Unused mean_squared_error imported from sklearn.metrics (unused-import)
train_pipeline.py:9:0: W0611: Unused r2_score imported from sklearn.metrics (unused-import)

-----------------------------------
Your code has been rated at 8.73/10
```

6. Go to the bikeshare_model directory and run formatting command



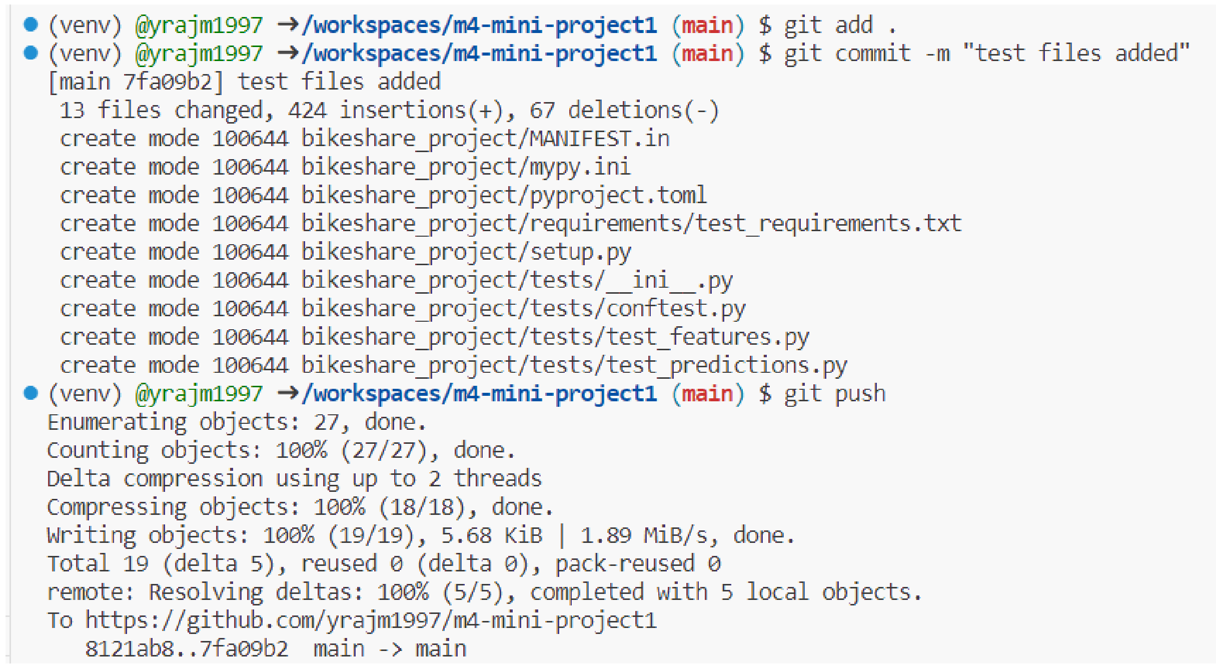
```
● (venv) @yrajm1997 →/workspaces/m4-mini-project1/bikeshare_project/bikeshare_model (main) $ black *.py
reformatted __init__.py
reformatted predict.py
reformatted train_pipeline.py
reformatted pipeline.py

All done! ✨ 🍰 ✨
4 files reformatted.
```

Step 7: Push the changes to the remote GitHub repository: (1 point)

7.1 On successful execution of the training , testing and formatting commands, push the new files and changes to the remote GitHub repository.

1. Delete the pickle file within the trained_models folder, as we want to generate it again during github actions workflow.

2. Perform git add, git commit, and git push to push your changes



```
● (venv) @yrajm1997 ➜/workspaces/m4-mini-project1 (main) $ git add .
● (venv) @yrajm1997 ➜/workspaces/m4-mini-project1 (main) $ git commit -m "test files added"
  [main 7fa09b2] test files added
   13 files changed, 424 insertions(+), 67 deletions(-)
   create mode 100644 bikeshare_project/MANIFEST.in
   create mode 100644 bikeshare_project/mypy.ini
   create mode 100644 bikeshare_project/pyproject.toml
   create mode 100644 bikeshare_project/requirements/test_requirements.txt
   create mode 100644 bikeshare_project/setup.py
   create mode 100644 bikeshare_project/tests/__ini__.py
   create mode 100644 bikeshare_project/tests/conftest.py
   create mode 100644 bikeshare_project/tests/test_features.py
   create mode 100644 bikeshare_project/tests/test_predictions.py
● (venv) @yrajm1997 ➜/workspaces/m4-mini-project1 (main) $ git push
  Enumerating objects: 27, done.
  Counting objects: 100% (27/27), done.
  Delta compression using up to 2 threads
  Compressing objects: 100% (18/18), done.
  Writing objects: 100% (19/19), 5.68 KiB | 1.89 MiB/s, done.
  Total 19 (delta 5), reused 0 (delta 0), pack-reused 0
  remote: Resolving deltas: 100% (5/5), completed with 5 local objects.
  To https://github.com/yrajm1997/m4-mini-project1
     8121ab8..7fa09b2  main -> main
```

Step 8: Create a GitHub Actions workflow to automate the steps for model training, testing, linting, and formatting: (2 points)

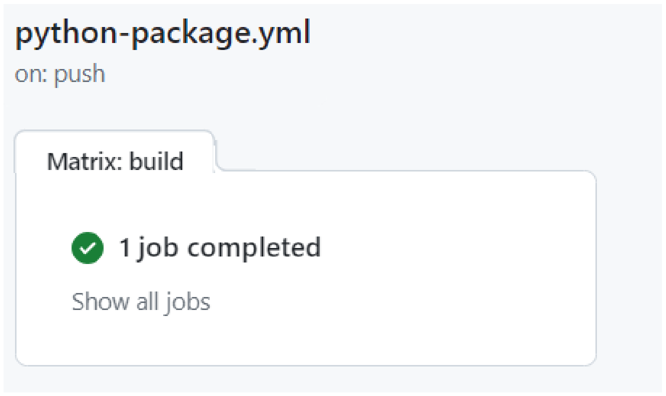
8.1 Create a GitHub Actions workflow to automate the steps for model training, testing, linting, and formatting.
8.2 Add below event triggers to the workflow:

- Run on push to main branch
- Run manually from the UI

1. On the GitHub repository page, go to *Actions tab > set up a workflow yourself*.
2. Add a .yml file with below content:

```yaml
name: Python app
on:
  push:
    branches: [ "main" ]
  pull_request:
    branches: [ "main" ]
  workflow_dispatch:
jobs:
  build:
    runs-on: ubuntu-latest
    strategy:
      fail-fast: false
      matrix:
        python-version: ["3.10"]
    steps:
    - name: Checkout step to clone repo
      uses: actions/checkout@v3
    - name: Set up Python ${{ matrix.python-version }}
      uses: actions/setup-python@v3
      with:
        python-version: ${{ matrix.python-version }}
    - name: Install dependencies
      run: |
        pip install -r bikeshare_project/requirements/test_requirements.txt
    - name: Train pipeline
      run: |
        python bikeshare_project/bikeshare_model/train_pipeline.py
    - name: Test with pytest
      run: |
        pytest
    - name: Format code with Black
      run: |
        black bikeshare_project/bikeshare_model/*.py
    - name: Lint with pylint
      run: |
        pylint --disable=R,C
bikeshare_project/bikeshare_model/{pipeline,train_pipeline,predict}.py
```

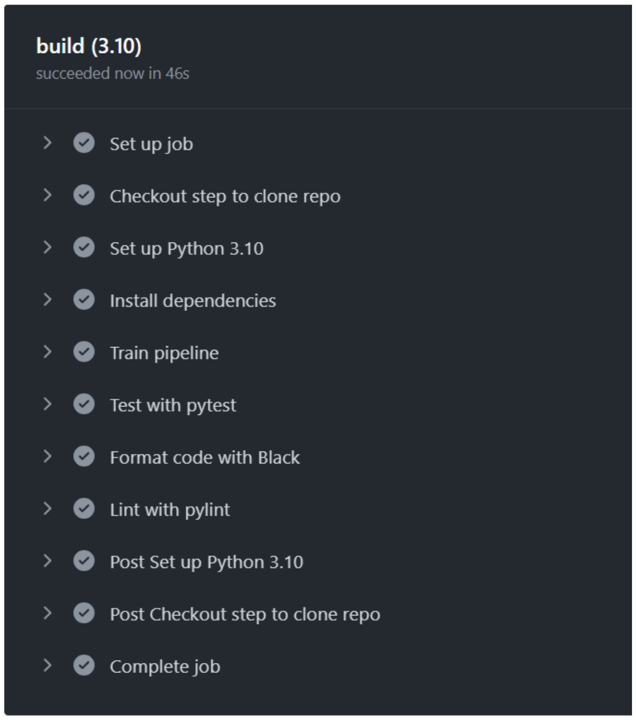
3. Check the workflow created under the Actions tab. The workflow should run fine.


python-package.yml
on: push

Matrix: build

✓ 1 job completed

Show all jobs



build (3.10)

succeeded now in 46s

- ✓ Set up job
- ✓ Checkout step to clone repo
- ✓ Set up Python 3.10
- ✓ Install dependencies
- ✓ Train pipeline
- ✓ Test with pytest
- ✓ Format code with Black
- ✓ Lint with pylint
- ✓ Post Set up Python 3.10
- ✓ Post Checkout step to clone repo
- ✓ Complete job