2025

4Geeks Academy: data science cohort 12

DAY 33: CI/CD

TODO

CI/CD

Testing, GitHub actions and workflows

DEPLOY

Deploy movie recommendation system web app to Render

TOPICS

01 CI/CD

02 TESTING

03 GITHUB WORKFLOW AUTOMATION

CI/CD

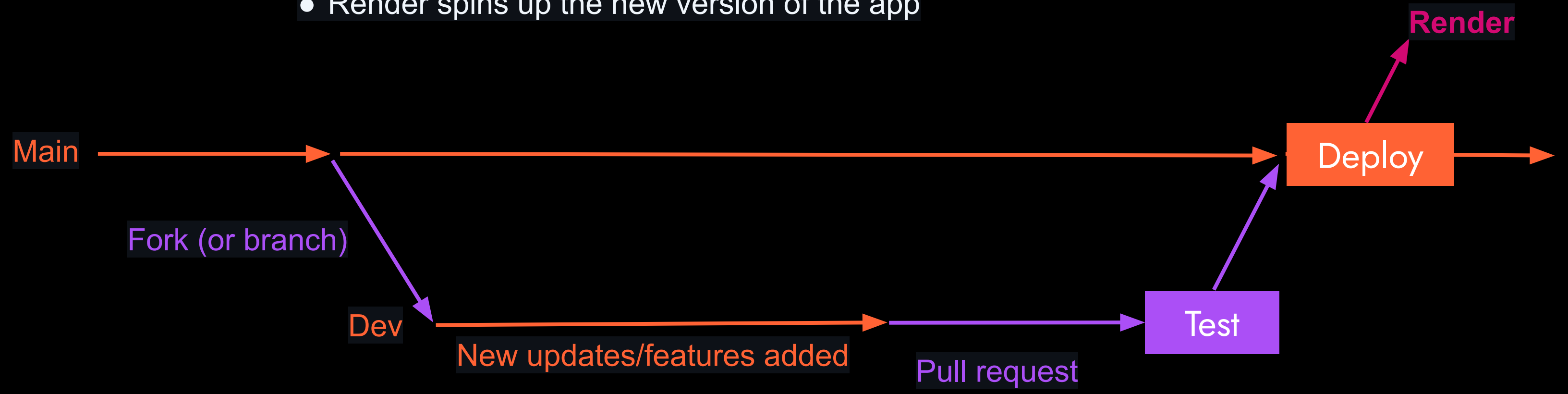
WHAT

- Stands for continuous integration and continuous delivery (and/or deployment)
- A set of tools and techniques to automate testing, building and delivering and or deploying software

HOW

Example workflow:

- A developer on the project submits a pull request containing updates to the main fork
- The pull request triggers automatics tests to make sure the code works as expected
- A human being sanity checks and merges or closes the pull request
- Merging the pull request triggers automatic deployment of the updated app
- Render spins up the new version of the app



TESTING

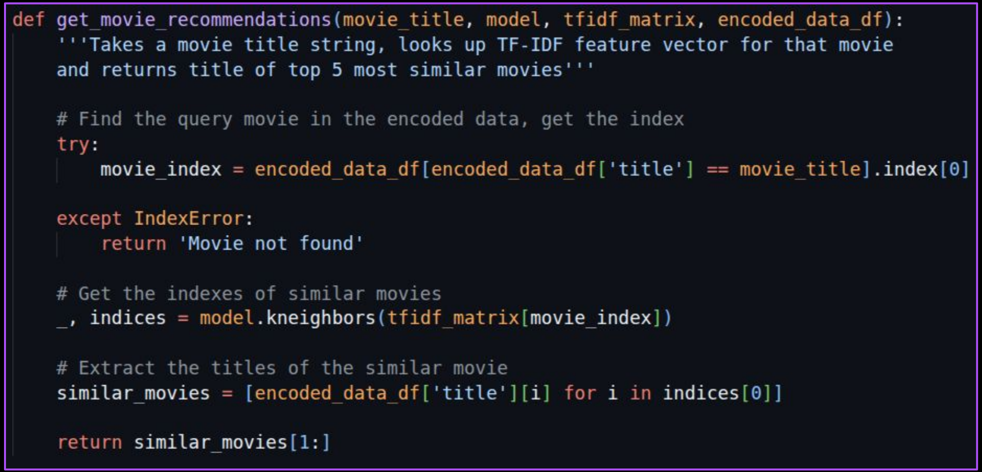
WHAT Python functions that run the project code and verify expected outputs. Can be automated.

HOW

- **Unittest**: Python’s standard library testing framework
- **PyTest:** small lightweight 3rd party testing framework

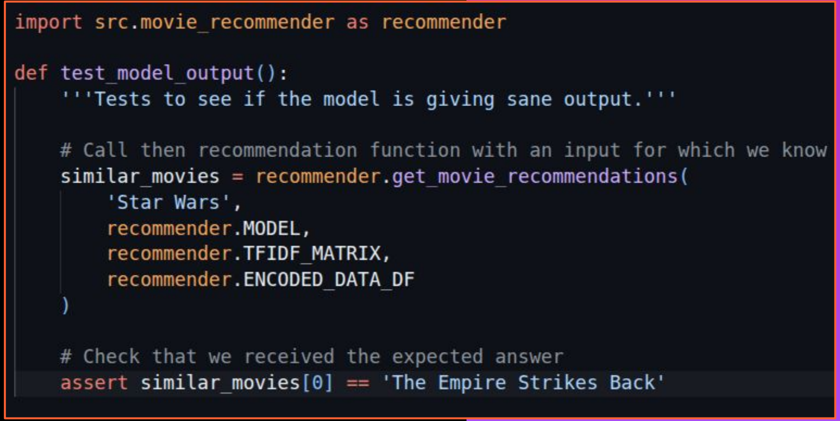
Testing uses test cases and ‘assert’ methods - you assert that output from a test case should (or should not be) something and the testing framework checks it.

FUNCTION



```python
def get_movie_recommendations(movie_title, model, tfidf_matrix, encoded_data_df):
    '''Takes a movie title string, looks up TF-IDF feature vector for that movie
    and returns title of top 5 most similar movies'''

    # Find the query movie in the encoded data, get the index
    try:
        movie_index = encoded_data_df[encoded_data_df['title'] == movie_title].index[0]

    except IndexError:
        return 'Movie not found'

    # Get the indexes of similar movies
    _, indices = model.kneighbors(tfidf_matrix[movie_index])

    # Extract the titles of the similar movie
    similar_movies = [encoded_data_df['title'][i] for i in indices[0]]

    return similar_movies[1:]
```

TEST



```python
import src.movie_recommender as recommender

def test_model_output():
    '''Tests to see if the model is giving sane output.'''

    # Call then recommendation function with an input for which we know
    similar_movies = recommender.get_movie_recommendations(
        'Star Wars',
        recommender.MODEL,
        recommender.TFIDF_MATRIX,
        recommender.ENCODED_DATA_DF
    )

    # Check that we received the expected answer
    assert similar_movies[0] == 'The Empire Strikes Back'
```

GITHUB WORKFLOW AUTOMATION

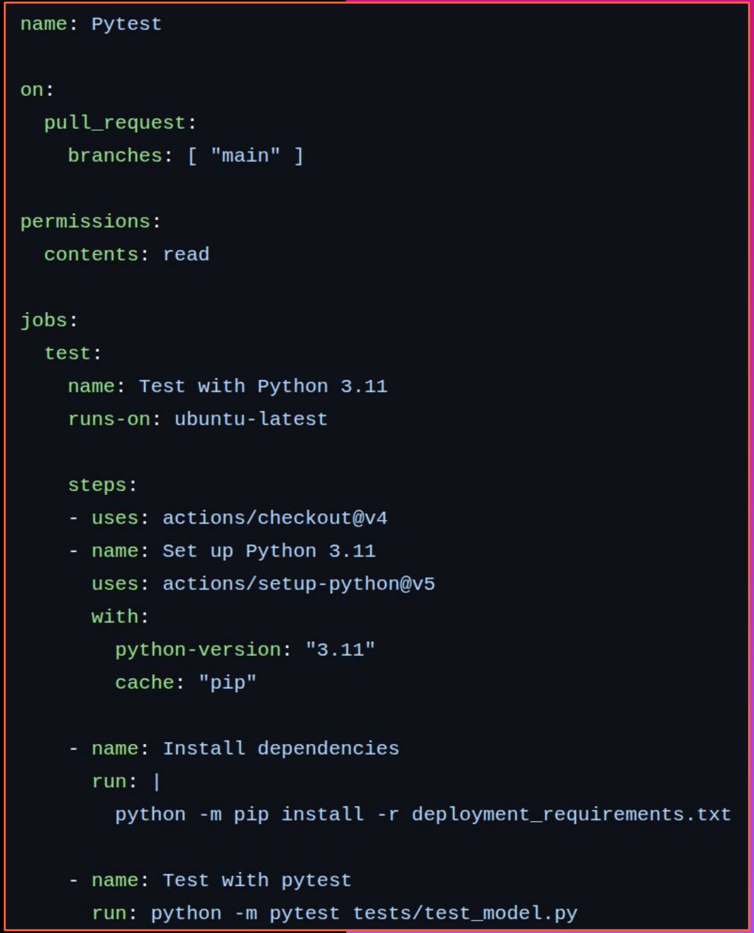
WHAT

GitHub has a set of tools built in to perform automated actions in response to specific events.

HOW

Defined via YML files in `.github/workflows/` directory

EX: run the test for the movie recommender project whenever a pull request is made to main



```
name: Pytest

on:
  pull_request:
    branches: [ "main" ]

permissions:
  contents: read

jobs:
  test:
    name: Test with Python 3.11
    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v4
    - name: Set up Python 3.11
      uses: actions/setup-python@v5
      with:
        python-version: "3.11"
        cache: "pip"

    - name: Install dependencies
      run: |
        python -m pip install -r deployment_requirements.txt

    - name: Test with pytest
      run: python -m pytest tests/test_model.py
```