Assignment 02

Deadline: 14-12-2025

Each student must submit the following:

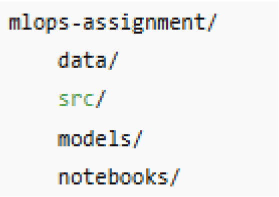
✓ **PDF/Word report** with:

- Explanation + Commands used
- **Terminal logs**
- **Screenshots** of outputs
- GitHub repository link (if any)
- AWS public endpoint URL

✓ **Zipped project folder** (excluding large files).

Task 1 — Project Setup + Version Control (Git + DVC)

1.1 Create Project Structure



```
mlops-assignment/
    data/
    src/
    models/
    notebooks/
```

Activity:

- Initialize git
- Add .gitignore
- Create a Python environment

Deliverables:

- Screenshot of folder structure
- Screenshot of git init output

1.2 Initialize DVC

Run:
dvc init
dvc remote add -d myremote ./dvcstore

Activity:

- Add dataset to data/
- Track it using DVC:

dvc add data/dataset.csv
git add data/.gitignore data/dataset.csv.dvc
git commit -m "Add dataset with DVC"

Deliverables:

- Screenshot of DVC commands
- Screenshot of .dvc file
- DVC status/output logs

1.3 Create a Basic Training Pipeline

Create src/train.py that:

- Loads dataset
- Trains a small model
- Saves model to models/model.pkl

Create DVC pipeline:

```
dvc run -n train_model \
  -d src/train.py -d data/dataset.csv \
  -o models/model.pkl \
  python src/train.py
```

Deliverables:

- Screenshot of successful DVC pipeline run
- dvc.yaml file screenshot

Task 2 — CI/CD Pipeline (GitHub Actions)

2.1 Create Workflow File

In .github/workflows/ci.yml:

Include steps:

- Setup Python
- Install requirements
- Run unit tests
- Run linting (flake8 or pylint)
- Verify training script runs

Deliverables:

- Screenshot of workflow file
- Screenshot of GitHub Actions successful run
- Include failing workflow screenshot if any

2.2 Add Tests

In tests/test_train.py, write tests for:

- Data loading
- Model training
- Shape validation

Deliverables:

- Screenshot of test results

Task 3 — Docker (Full Assignment)

3.1 Create Dockerfile

Container should:

- Install dependencies
- Copy project files
- Run training script or API

Example:

```
FROM python:3.10
WORKDIR /app
COPY . .
RUN pip install -r requirements.txt
CMD ["python", "src/train.py"]
```

Activity:

Build and run the container:

```
docker build -t mlops-app .
docker run mlops-app
```

Deliverables:

- Dockerfile screenshot
- Build logs
- Running container screenshot

3.2 Push Docker Image to Docker Hub

Commands:

```
docker tag mlops-app yourusername/mlops-app:v1
docker push yourusername/mlops-app:v1
```

Deliverables:

- Screenshot of Docker Hub repository page
- Screenshot of push logs

Task 4 — Airflow Pipeline

4.1 Install Airflow Locally (Docker Compose)

Use Airflow official docker-compose file.

Activity:

- Create DAG train_pipeline.py that:
    - Loads data
    - Trains model
    - Saves trained model
    - Logs results

Deliverables:

- Screenshot of Airflow UI
- Screenshot of DAG graph
- Screenshot of successful job run

Task 5 — RESTful API (FastAPI/Flask)

5.1 Build ML Inference API

Create api/main.py:

Endpoints:

- /predict → returns model prediction
- /health → returns status

Run:

```
uvicorn api.main:app --reload
```

Deliverables:

- Screenshot of API running
- Screenshot of testing using Postman/cURL
- Sample prediction outputs

5.2 Containerize the API

Modify Dockerfile:

```
CMD ["uvicorn", "api.main:app", "--host", "0.0.0.0", "--port", "8000"]
```

Run:

```
docker build -t mlops-api .
docker run -p 8000:8000 mlops-api
```

Deliverables:

- API running in Docker screenshot

Task 6 — AWS EC2 + S3 Deployment (Free Tier)

6.1 Create AWS S3 Bucket

Activity:

- Create bucket
- Upload dataset
- Configure permissions
- Generate bucket URL

Deliverables:

- Screenshot of S3 bucket
- Screenshot of uploaded data

6.2 Launch EC2 Instance (Free Tier)

- Choose **Ubuntu 22.04**
- Open port **8000** for API
- Install dependencies

Run:

```
sudo apt update
sudo apt install docker.io python3-pip
```

Deliverables:

- Screenshot of EC2 instance dashboard

- Screenshot of instance SSH terminal

6.3 Deploy API Using Docker on EC2

Commands:

```
docker pull yourusername/mlops-api:v1
docker run -d -p 8000:8000 yourusername/mlops-api:v1
```

Deliverables:

- Screenshot of logs
- Screenshot of running container
- Public endpoint test screenshot (browser/Postman)

Task 7 — Final Deliverables and Reflection

Students must submit:

1. GitHub repository link
2. Docker Hub link
3. Airflow DAG screenshot
4. EC2 Public API URL
5. DVC pipeline screenshot
6. CI/CD workflow screenshot
7. Report with:
   - Problems faced
   - Fixes applied
   - Learning summary