Python & Machine Learning Project Guidelines

1. Project Objective

Build a **production-ready Machine Learning API service** using Python that:

- Trains or loads an ML model
- Exposes prediction endpoints via REST APIs
- Is fully containerized using Docker
- Is automatically tested, validated, and deployed through CI/CD

This project is evaluated on **engineering quality**, not just model accuracy.

2. Mandatory Deliverables

Each student **must submit ALL of the following**:

1. GitHub Repository URL

Must include:

- Source code
- Tests
- Dockerfile
- GitHub Actions workflow
- README.md with setup instructions

2. Docker Hub Image URL

Example format:

```
https://hub.docker.com/r/<username>/<image-name>
```

Important:

- Docker image **MUST be pushed automatically using GitHub Actions**
- Manual Docker uploads are NOT allowed

- GitHub Actions logs must show successful build + push

3. curl Commands to Test APIs

Students must provide working curl commands such as:

```
curl -X POST http://localhost:8000/predict \
-H "Content-Type: application/json" \
-d '{"feature1": 10, "feature2": 5}'
```

At minimum:

- One health endpoint
- One prediction endpoint

3. Technical Requirements

API Framework

Use one of:

- FastAPI (preferred)
- Flask

ML Model

Can be:

- Classification or Regression
- scikit-learn / XGBoost / LightGBM / CatBoost

Must include:

- Feature preprocessing
- Model persistence (joblib / pickle)
- Prediction endpoint

4. CI/CD (MANDATORY)

GitHub Actions pipeline must automatically perform:

Step 1 — PEP8 Formatting Check

Using:

```
flake8
```

Pipeline must fail if violations exist.

Step 2 — Bandit Security Scan

Using:

```
bandit -r .
```

No HIGH severity issues allowed.

Step 3 — Unit Testing + Coverage

Using:

```
pytest
pytest-cov
```

Coverage requirement:

```
≥ 80%
```

Pipeline must fail if coverage is below threshold.

Step 4 — Docker Build + Push

Pipeline must:

1. Build Docker image
2. Login to Docker Hub using GitHub Secrets
3. Push image automatically

Example tools:

- docker/login-action
- docker/build-push-action

Manual Docker builds = automatic failure.

5. Docker Requirements

Docker image must:

- Start API server automatically
- Expose correct port
- Contain all dependencies
- Be runnable with:

```
docker run -p 8000:8000 <image>
```

6. Code Quality Standards

Project must follow:

PEP8 Compliance

Modular structure

Clear separation:

- app/
- model/
- tests/
- Dockerfile
- requirements.txt

7. Testing Requirements

Must include:

- Unit tests for API routes
- Unit tests for model logic
- Edge case handling

Minimum:

```
80% coverage
```

8. README.md Must Contain

Students must document:

1. Project overview
2. How to run locally
3. How to build Docker image
4. curl commands
5. Docker Hub URL
6. CI/CD explanation

9. Evaluation Criteria

Category	Weight
CI/CD pipeline	30%
Code quality + PEP8	20%
Tests + Coverage	20%
Dockerization	15%
API design	10%
Documentation	5%

Project will be disqualified If

- Docker image pushed manually
- No GitHub Actions pipeline

- Coverage < 80%
- Bandit HIGH severity issues
- Missing curl commands
- Missing Docker Hub URL