

MLOps Assignment I — End-to-End ML Model Development, CI/CD, and Production Deployment

1. Objective

Build a machine learning classifier for heart disease prediction and deploy it as a cloud-ready, monitored API using modern MLOps best practices.

2. Demo Video & Repository Link

- **Demo Video:** [Pipeline Demo on YouTube](#) - Complete end-to-end pipeline demonstration
- **GitHub Repository:** github.com/chaitudevi/Heart-Disease-Classification-2

3. Dataset (Heart Disease UCI)

- **Source:** UCI Machine Learning Repository
- **Data acquisition:**
 - Script: `src/data/download_data.py`
 - Raw data folder: `data/raw/`

Screenshot(s)

```
darshan@Darshans-MacBook-Air Heart-Disease-Classification-2 % python src/data/download_data.py
Dataset downloaded to /Users/darshan/Developer/repos/Heart-Disease-Classification-2/data/raw/heart_disease.csv
Column headers added successfully
```

4. Data Acquisition, Cleaning, and EDA (Task 1 — 5 marks)

4.1 Data Cleaning & Preprocessing

- Missing value handling, encoding, transformations
- Implementation: `src/data/preprocess.py`

4.2 EDA

- Notebook: `notebooks/01_eda.ipynb`
- Visuals: class balance, correlation heatmap, distributions

Screenshot(s)



5. Feature Engineering & Model Development (Task 2 — 8 marks)

5.1 Feature Pipeline

- Implementation: `src/features/feature_pipeline.py`

5.2 Models

- Logistic Regression
- Random Forest

5.3 Evaluation

- Cross-validation metrics: accuracy, precision, recall, ROC-AUC
- Training entrypoint: `src/models/train.py`

Screenshot(s)

```
darshan@Darshans-MacBook-Air Heart-Disease-Classification-2 % python src/models/train.py
/Users/darshan/.pyenv/versions/3.10.18/lib/python3.10/site-packages/pydantic/_internal/_fields.py:149: UserWarning: Field "model_name" has conflict with protected namespace "model_"
.

You may be able to resolve this warning by setting `model_config['protected_namespaces'] = ()`.
    warnings.warn(
/Users/darshan/.pyenv/versions/3.10.18/lib/python3.10/site-packages/mlflow/tracking/_tracking_service/utils.py:178: FutureWarning: The filesystem tracking backend (e.g., './mlruns') will be deprecated in February 2026. Consider transitioning to a database backend (e.g., 'sqlite:///mlflow.db') to take advantage of the latest MLflow features. See https://github.com/mlflow/mlflow/issues/18534 for more details and migration guidance. For migrating existing data, https://github.com/mlflow/mlflow-export-import can be used.
    return FileStore(store_uri, store_uri)
[2026-01-03 13:54:13,265] INFO | src.features.feature_pipeline | Building full feature pipeline
[2026-01-03 13:54:13,265] INFO | src.features.feature_pipeline | Creating ColumnTransformer
[2026-01-03 13:54:13,265] INFO | src.features.feature_pipeline | Building numeric feature pipeline
[2026-01-03 13:54:13,265] INFO | src.features.feature_pipeline | Building categorical feature pipeline
[2026-01-03 13:54:13,268] INFO | src.features.feature_pipeline | Creating engineered features
[2026-01-03 13:54:13,277] INFO | src.features.feature_pipeline | Creating engineered features
/Users/darshan/.pyenv/versions/3.10.18/lib/python3.10/site-packages/sklearn/utils/extmath.py:203: RuntimeWarning: divide by zero encountered in matmul
    ret = a @ b
/Users/darshan/.pyenv/versions/3.10.18/lib/python3.10/site-packages/sklearn/utils/extmath.py:203: RuntimeWarning: overflow encountered in matmul
    ret = a @ b
/Users/darshan/.pyenv/versions/3.10.18/lib/python3.10/site-packages/sklearn/utils/extmath.py:203: RuntimeWarning: invalid value encountered in matmul
    ret = a @ b
[2026-01-03 13:54:13,282] INFO | src.features.feature_pipeline | Creating engineered features
/Users/darshan/.pyenv/versions/3.10.18/lib/python3.10/site-packages/sklearn/utils/extmath.py:203: RuntimeWarning: divide by zero encountered in matmul
    ret = a @ b
/Users/darshan/.pyenv/versions/3.10.18/lib/python3.10/site-packages/sklearn/utils/extmath.res
[2026-01-03 13:54:13,543] INFO | src.features.feature_pipeline | Creating engineered features
[2026-01-03 13:54:13,559] INFO | src.features.feature_pipeline | Creating engineered features
[2026-01-03 13:54:13,576] INFO | src.features.feature_pipeline | Creating engineered features
[2026-01-03 13:54:13,662] INFO | src.features.feature_pipeline | Creating engineered features
[2026-01-03 13:54:13,680] INFO | src.features.feature_pipeline | Creating engineered features

Logistic Regression
accuracy: 0.8429
precision: 0.8620
recall: 0.7838
roc_auc: 0.9056

Random Forest
accuracy: 0.7933
precision: 0.8024
recall: 0.7297
roc_auc: 0.8853
[2026-01-03 13:54:13,700] INFO | src.features.feature_pipeline | Building full feature pipeline
[2026-01-03 13:54:13,700] INFO | src.features.feature_pipeline | Creating ColumnTransformer
```

```
r
[2026-01-03 13:54:13,700] INFO | src.features.feature_pipeline | Building numeric feature
pipeline
[2026-01-03 13:54:13,700] INFO | src.features.feature_pipeline | Building categorical feat
ure pipeline
[2026-01-03 13:54:13,701] INFO | src.features.feature_pipeline | Creating engineered featu
res
Best model selected: Logistic Regression
[2026-01-03 13:54:13,707] INFO | src.features.feature_pipeline | Creating engineered featu
res
```

6. Experiment Tracking (Task 3 — 5 marks)

- Tool: MLflow
- Logged:
 - Parameters
 - Metrics
 - Artifacts (ROC curve, confusion matrix, reports)

Screenshot(s)

The screenshot shows the MLflow UI at the URL `http://localhost:5001/#/experiments`. The interface has a sidebar with options for Home, Experiments (which is selected), Models, and Prompts. The main area is titled "Experiments" and displays a table of existing experiments. The table columns are Name, Time created, Last modified, Description, and Tags. One experiment is listed: "Heart-Disease-Classification-2" was created on 12/30/2025, 08:18:51 PM, last modified on 12/30/2025, 08:18:51 PM, and has no description or tags.

7. Model Packaging & Reproducibility (Task 4 — 7 marks)

- Model bundle saved at: `artifacts/model.pkl`
- Reproducible environment: `requirements.txt`
- Preprocessing/feature transformations are part of the saved sklearn `Pipeline`

Screenshot(s)

Heart-Disease-Classification-2 Machine learning

Runs

| Run Name | Created | Dataset | Duration | Source | Models |
|---------------------|------------|---------|----------|----------|---------|
| Best_Model | 3 days ago | - | 1.4s | train.py | sklearn |
| Random Forest | 3 days ago | - | 412ms | train.py | - |
| Logistic Regression | 3 days ago | - | 39ms | train.py | - |

Run details

Run ID: 1a122be60c944d4288dcff2986d6943 e3dc5b0e30634af5adca19d3c23d3f52

Run Name: Logistic Regression Random Forest

Start Time: 12/30/2025, 08:18:51 PM 12/30/2025, 08:18:51 PM

End Time: 12/30/2025, 08:18:51 PM 12/30/2025, 08:18:51 PM

Duration: 39ms 412ms

Parameters

Show diff only

| model_type | Logistic Regression | Random Forest |
|------------|---------------------|---------------|
|------------|---------------------|---------------|

Metrics

Show diff only

| accuracy | 0.845 | 0.825 |
|-----------|-------|-------|
| precision | 0.846 | 0.825 |
| recall | 0.813 | 0.784 |
| roc_auc | 0.911 | 0.909 |

Artifacts

Best_Model

- model
- metadata
- MLmodel
- conda.yaml
- model.pkl
- python_env.yaml
- requirements.txt
- model.pkl
- roc_curve.png

```
model/MLmodel 527B
Path: file:///Users/darshan/Developer/repos/Heart-Disease-Classification-2/mlruns/236215858674366643/e1a6465e5a484e7cae4053a7aa22dbcb/artifacts/model/...
artifact_path: model
flavors:
python_function:
env:
  conda: conda.yaml
  virtualenv: python_env.yaml
  loader_module: mlflow.sklearn
  model_path: model.pkl
  predict_fn: predict
  python_version: 3.10.18
sklearn:
  code: null
  pickled_model: model.pkl
  serialization_format: cloudpickle
  sklearn_version: 1.4.2
mlflow_version: 2.12.1
model_size_bytes: 4209
model_uuid: eadca141a4724a54ba175c73edf3b663
run_id: e1a6465e5a484e7cae4053a7aa22dbcb
utc_time_created: '2025-12-30 14:48:51.893519'
```

8. CI/CD Pipeline & Automated Testing (Task 5 — 8 marks)

- Tests under: `tests`
- CI workflow: `.github/workflows/github_actions.yaml`
- Stages: lint, test, train

Screenshot(s)

The screenshot shows the GitHub Actions CI Pipeline for the repository "Heart-Disease-Classification-2". A pull request titled "Fix github action yaml #20" has triggered a pipeline run. The run was triggered via push last week by user "mlwithak" and pushed to branch "da20229_fix/train_ci". The status is "Success" with a total duration of "1m 33s" and one artifact produced.

Artifacts:

| Name | Size | Digest |
|----------------------------|---------|--|
| <code>pytest-report</code> | 9.29 KB | sha256:8756ec0477f4660718db3c358b3a02832d8a... |

The screenshot shows the test logs for the "test" job in the CI pipeline. The job succeeded last week in 49s. The logs detail the execution of unit tests, showing warnings related to deprecated parsing logic and matplotlib fontconfig patterns. The logs conclude with 10 passed tests and 14 warnings in 4.02s.

```

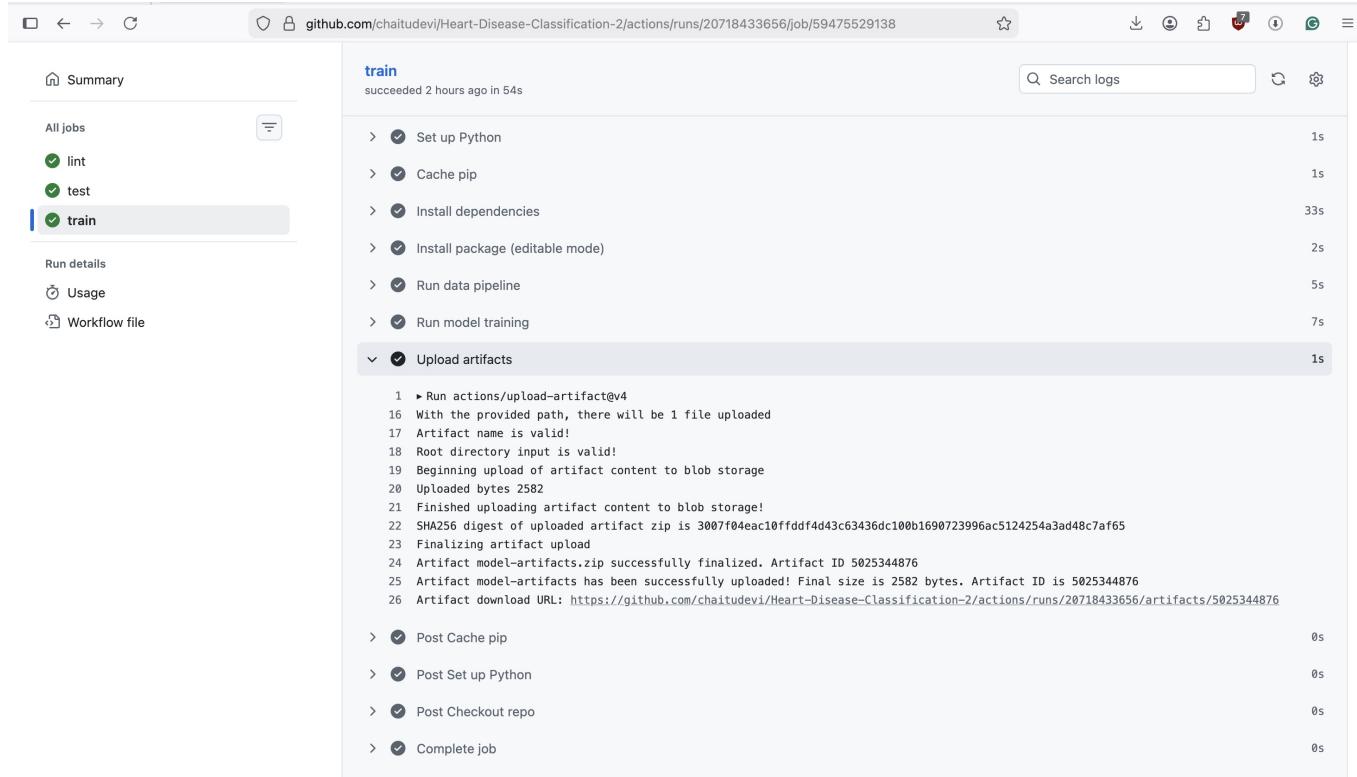
test
succeeded last week in 49s

Run unit tests
17 tests/data/test_download_data.py . [ 5%]
18 tests/data/test_ed_outputs.py . [ 11%]
19 tests/data/test_load_data.py . [ 16%]
20 tests/data/test_reprocess.py .... [ 38%]
21 tests/features/test_features.py .... [ 66%]
22 tests/models/test_model.py .. [ 77%]
23 tests/utils/test_logger.py .... [100%]

=====
25 ===== warnings summary =====
26 .../.../.../.../.../opt/hostedtoolcache/Python/3.10.19/x64/lib/python3.10/site-packages/matplotlib/_fontconfig_pattern.py:88
27 .../.../.../.../.../opt/hostedtoolcache/Python/3.10.19/x64/lib/python3.10/site-packages/matplotlib/_fontconfig_pattern.py:88
28 .../.../.../.../.../opt/hostedtoolcache/Python/3.10.19/x64/lib/python3.10/site-packages/matplotlib/_fontconfig_pattern.py:88
29 .../.../.../.../.../opt/hostedtoolcache/Python/3.10.19/x64/lib/python3.10/site-packages/matplotlib/_fontconfig_pattern.py:88
30 .../.../.../.../.../opt/hostedtoolcache/Python/3.10.19/x64/lib/python3.10/site-packages/matplotlib/_fontconfig_pattern.py:88
31 .../.../.../.../.../opt/hostedtoolcache/Python/3.10.19/x64/lib/python3.10/site-packages/matplotlib/_fontconfig_pattern.py:88
32 .../.../.../.../.../opt/hostedtoolcache/Python/3.10.19/x64/lib/python3.10/site-packages/matplotlib/_fontconfig_pattern.py:88
33 .../.../.../.../.../opt/hostedtoolcache/Python/3.10.19/x64/lib/python3.10/site-packages/matplotlib/_fontconfig_pattern.py:88
34 .../.../.../.../.../opt/hostedtoolcache/Python/3.10.19/x64/lib/python3.10/site-packages/matplotlib/_fontconfig_pattern.py:88
35 .../.../.../.../.../opt/hostedtoolcache/Python/3.10.19/x64/lib/python3.10/site-packages/matplotlib/_fontconfig_pattern.py:92
36 .../.../.../.../.../opt/hostedtoolcache/Python/3.10.19/x64/lib/python3.10/site-packages/matplotlib/_fontconfig_pattern.py:92
37 .../.../.../.../.../opt/hostedtoolcache/Python/3.10.19/x64/lib/python3.10/site-packages/matplotlib/_fontconfig_pattern.py:92
38 .../.../.../.../.../opt/hostedtoolcache/Python/3.10.19/x64/lib/python3.10/site-packages/matplotlib/_fontconfig_pattern.py:92
39 .../.../.../.../.../opt/hostedtoolcache/Python/3.10.19/x64/lib/python3.10/site-packages/matplotlib/_fontconfig_pattern.py:92
40 .../.../.../.../.../opt/hostedtoolcache/Python/3.10.19/x64/lib/python3.10/site-packages/matplotlib/_fontconfig_pattern.py:92
41 .../.../.../.../.../opt/hostedtoolcache/Python/3.10.19/x64/lib/python3.10/site-packages/matplotlib/_fontconfig_pattern.py:92: DeprecationWarning: 'resetCache' deprecated - use 'reset_cache'
42 .../.../.../.../.../parser.resetCache()
43
44 .../.../.../.../.../opt/hostedtoolcache/Python/3.10.19/x64/lib/python3.10/site-packages/matplotlib/_mathtext.py:45
45 .../.../.../.../.../opt/hostedtoolcache/Python/3.10.19/x64/lib/python3.10/site-packages/matplotlib/_mathtext.py:45: DeprecationWarning: 'enablePackrat' deprecated - use 'enable_packrat'
46 .../.../.../.../.../ParserElement.enablePackrat()
47
48 tests/data/test_reprocess.py::test_clean_data_replaces_missing
49 .../home/runner/work/Heart-Disease-Classification-2/Heart-Disease-Classification-2/src/data/preprocess.py:13: FutureWarning: Downcasting behavior in 'replace' is deprecated and will be removed in a future version. To retain the old behavior, explicitly call 'result.infer_objects(copy=False)'. To opt-in to the future behavior, set pd.set_option('future.no_silent_downcasting', True)'
50 .../df.replace(MISSING_MARKERS, np.nan)
51
52 -- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
53 - Generated html report: file:///home/runner/work/Heart-Disease-Classification-2/Heart-Disease-Classification-2/reports/pytest_report.html -
54 ====== 10 passed, 14 warnings in 4.02s ======
```

Actions:

- Upload test report
- Post Cache pip
- Post Set up Python



The screenshot shows the GitHub Actions logs for a 'train' job. The job summary indicates it succeeded 2 hours ago in 54s. The logs detail the steps taken:

- Set up Python (1s)
- Cache pip (1s)
- Install dependencies (33s)
- Install package (editable mode) (2s)
- Run data pipeline (5s)
- Run model training (7s)
- Upload artifacts (1s):
 - Run actions/upload-artifact@v4 (1s)
 - With the provided path, there will be 1 file uploaded (1s)
 - Artifact name is valid! (1s)
 - Root directory input is valid! (1s)
 - Beginning upload of artifact content to blob storage (1s)
 - Uploaded bytes 2582 (1s)
 - Finished uploading artifact content to blob storage! (1s)
 - SHA256 digest of uploaded artifact zip is 3007f04eac10ffdd4d43c63436dc100b1690723996ac5124254a3ad48c7af65 (1s)
 - Finalizing artifact upload (1s)
 - Artifact model-artifacts.zip successfully finalized. Artifact ID 5025344876 (1s)
 - Artifact model-artifacts has been successfully uploaded! Final size is 2582 bytes. Artifact ID is 5025344876 (1s)
 - Artifact download URL: <https://github.com/chaitudevi/Heart-Disease-Classification-2/actions/runs/20718433656/artifacts/5025344876> (1s)
- Post Cache pip (0s)
- Post Set up Python (0s)
- Post Checkout repo (0s)
- Complete job (0s)

9. Model Containerization (Task 6 — 5 marks)

- Dockerfile: [Dockerfile](#)
- API:
 - `/predict` accepts JSON, returns prediction + confidence
 - `/metrics` exposes Prometheus metrics

Screenshot(s)

```
darshan@Darshans-MacBook-Air Heart-Disease-Classification-2 % docker build -t heart-disease-api:latest .
[+] Building 2.5s (11/11) FINISHED
    => [internal] load build definition from Dockerfile
    => => transferring dockerfile: 826B
    => [internal] load metadata for docker.io/library/python:3.10-slim
    => [internal] load .dockerignore
    => => transferring context: 207B
    => [1/6] FROM docker.io/library/python:3.10-slim@sha256:7b68a5fa7cf0d20b4cedb1dc9a
    => => resolve docker.io/library/python:3.10-slim@sha256:7b68a5fa7cf0d20b4cedb1dc9a
    => [internal] load build context
    => => transferring context: 1.09MB
    => CACHED [2/6] WORKDIR /app
    => CACHED [3/6] RUN apt-get update && apt-get install -y --no-install-recommends
    => CACHED [4/6] COPY requirements.txt ./requirements.txt
    => CACHED [5/6] RUN pip install --no-cache-dir -r requirements.txt
    => [6/6] COPY .
    => exporting to image
    => => exporting layers
    => => exporting manifest sha256:9e558d454705301252952463e92faeee02e025a8eed006548a
    => => exporting config sha256:351153723d5ba02edbd88e8fcfa055982324e9f1b23aff0b9ca9b
    => => exporting attestation manifest sha256:658ff0f73f69b0ed51642dbe9168e6a75f20ec
    => => exporting manifest list sha256:6bc4d960591420c677ee391fa46bd4a36fb4193421f89
    => => naming to docker.io/library/heart-disease-api:latest
    => => unpacking to docker.io/library/heart-disease-api:latest
    => => 0.0s
```

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/p51azyqgbz6g5tu3rnlpv1e03

```
darshan@Darshans-MacBook-Air Heart-Disease-Classification-2 % docker run --rm -p 8000:8000 -v $(pwd)/artifacts:/app/artifacts heart-disease-api:latest
INFO: Started server process [1]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
[2026-01-03 08:46:01,542] INFO | src.api.app | Handled GET /health -> 200 in 0.013s
INFO: 127.0.0.1:45314 - "GET /health HTTP/1.1" 200 OK
```

```
darshan@Darshans-MacBook-Air Heart-Disease-Classification-2 % curl -X POST http://localhost:8000/predict -H "Content-Type: application/json" --data @sample_data/sample_request.json
{"prediction":0,"confidence":0.8095673592559162}
```

10. Production Deployment (Task 7 — 7 marks)

You deployed on **Docker Desktop Kubernetes** using manifests.

- Manifest: [k8s/deployment.yaml](#)
- Service exposure: LoadBalancer/NodePort

Screenshot(s)

```
darshan@Darshans-MacBook-Air Heart-Disease-Classification-2 % kubectl apply -f k8s/deployment.yaml
deployment.apps/heart-disease-api unchanged
service/heart-disease-api unchanged
darshan@Darshans-MacBook-Air Heart-Disease-Classification-2 %

● darshan@Darshans-MacBook-Air Heart-Disease-Classification-2 % kubectl get svc heart-disease-api
  NAME           TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)      AGE
  heart-disease-api   LoadBalancer   10.103.110.219   localhost   80:31677/TCP   3d2h
○ darshan@Darshans-MacBook-Air Heart-Disease-Classification-2 %

● darshan@Darshans-MacBook-Air Heart-Disease-Classification-2 % kubectl get pods
  NAME           READY   STATUS    RESTARTS   AGE
  heart-disease-api-867c4f967b-wb2xl   1/1     Running   0          3d2h
○ darshan@Darshans-MacBook-Air Heart-Disease-Classification-2 %
```

The screenshot shows the Docker Desktop Kubernetes interface. At the top, there's a navigation bar with icons for recent projects, Docker Hub, and Docker Compose. The main area is titled "Kubernetes" with a "Give feedback" link. A dropdown menu for "Namespace" is set to "default". On the left, a sidebar has icons for Home, Projects, Services, Deployments, and Events.

Cluster

| Cluster | Cluster type | Nodes | Kubernetes version |
|---------|--------------|-------|--------------------|
| Active | kubeadm | 1 | v1.34.1 |

Deployments

| Name | Status | Pods |
|-------------------|-----------|------|
| heart-disease-api | Available | 1/1 |

Pods

| Name | Status |
|------------------------------------|---------|
| heart-disease-api-867c4f967b-wb2xl | Running |

Nodes

| Name | Status |
|----------------|--------|
| docker-desktop | Ready |

Services

| Name | Cluster IP | Ports |
|-------------------|----------------|---------|
| heart-disease-api | 10.103.110.219 | 80/TCP |
| kubernetes | 10.96.0.1 | 443/TCP |

11. Monitoring & Logging (Task 8 — 3 marks)

11.1 API Logging

- Middleware request logging in: `src/api/app.py`

11.2 Metrics + Dashboards

- Metrics endpoint: `/metrics`
- Prometheus config: `monitoring/prometheus.yml`
- Grafana provisioning + dashboard:
 - `monitoring/grafana/provisioning/`
 - `monitoring/grafana/dashboards/heart-disease-api-dashboard.json`

Screenshot(s)

Screenshot of the Prometheus UI showing target health for the 'heart-disease-api' job. The endpoint is `http://host.docker.internal:80/metrics` and it is marked as UP.

Screenshot of Grafana displaying various metrics for the Heart Disease API. The top section shows Request rate (req/s), Latency p50/p95/p99 (seconds), HTTP error rate (4xx/5xx req/s), and Prediction confidence p50. The bottom section shows Top 5 endpoints by request rate and Latency p95 by endpoint (seconds). All metrics show a sharp increase starting around 13:30 UTC.

Screenshot of the terminal showing log output from the application. The logs show multiple INFO messages indicating successful handling of GET requests for metrics, health, and favicon endpoints. The log entries are timestamped from 2026-01-03 08:51:34 to 08:51:59.

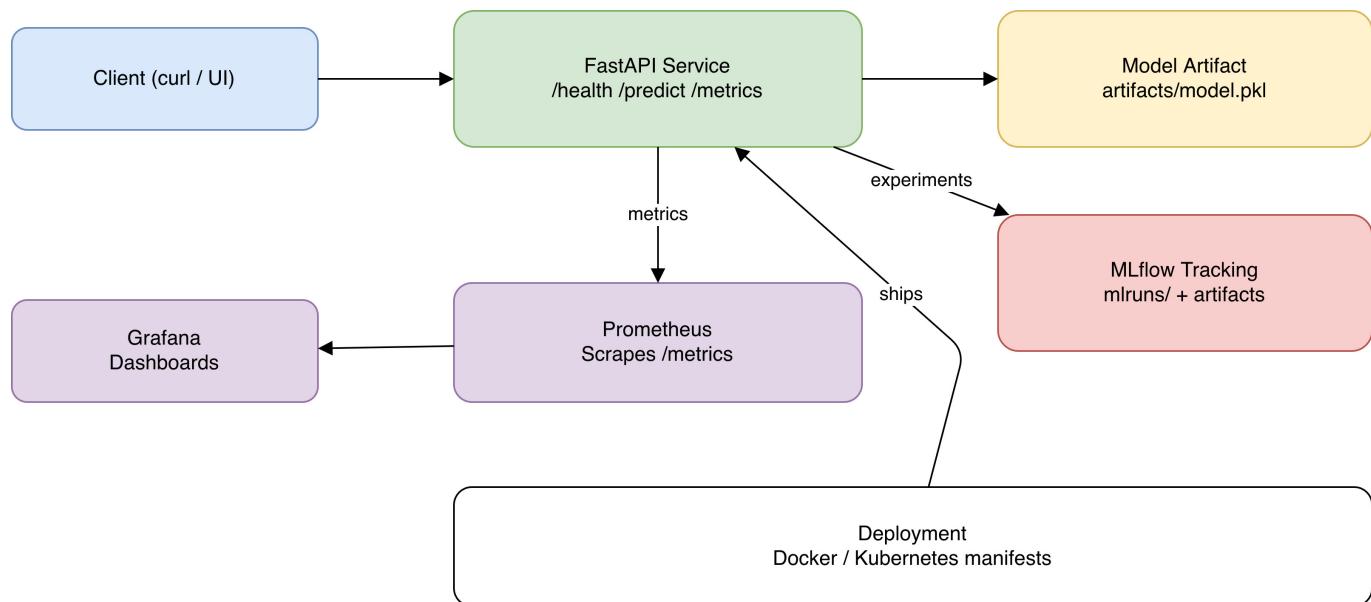
12. System Architecture Diagram

- Draw.io XML: [reports/system_architecture.xml](#)

- Architecture flow:

```
graph TD
    A[Client / UI] --> B[FastAPI Service]
    B --> C[Model Artifact joblib]
    B --> D[Prometheus Metrics]
    B --> E[Logging]
    E --> F[Stdout / Aggregator]
    D --> G[Grafana / Prometheus]
```

Screenshot(s)



13. How to Reproduce (Production-Readiness)

13.1 From clean environment

```
# 1. Clone and setup
git clone https://github.com/chaitudevi/Heart-Disease-Classification-2
cd Heart-Disease-Classification-2
python -m venv venv && source venv/bin/activate
pip install -r requirements.txt

# 2. Data pipeline
python -m src.data.download_data
python -m src.data.load_data

# 3. Train model (generates artifacts/model.pkl)
python -m src.models.train
# 4. Run API locally
uvicorn src.api.app:app --host 0.0.0.0 --port 8000

# 5. Test API
curl -X POST http://localhost:8000/predict -H "Content-Type: application/json" \
```

```
-d @sample_data/sample_request.json

# 6. Run tests
./venv/bin/python -m pytest tests -v

# 7. Build container
docker build -t heart-disease-api .

# 8. Deploy to Kubernetes
kubectl apply -f k8s/deployment.yaml

# 9. Start monitoring stack
docker-compose -f docker-compose.monitoring.yml up -d

# 10. View MLflow experiments
mlflow ui --host 0.0.0.0 --port 5000
```

13.2 Key Config Files

- `monitoring/prometheus.yaml`: Prometheus scrape config
- `monitoring/grafana/provisioning/`: Grafana auto-provision
- `k8s/deployment.yaml`: Kubernetes manifests
- `.github/workflows/github_actions.yaml`: CI/CD pipeline
- `src/models/train.py`: Training script with MLflow logging
- `src/api/app.py`: FastAPI app with metrics and logging

13.3 Ports & Endpoints

- API: `http://localhost:8000` (or k8s service IP)
- Health: `http://localhost:8000/health`
- Metrics: `http://localhost:8000/metrics`
- Prometheus: `http://localhost:9090`
- Grafana: `http://localhost:3000` (admin/admin123)
- MLflow: `http://localhost:5000`

14. Appendix: Evidence Checklist

- CI run screenshots
- Deployment screenshots
- Grafana dashboard screenshot
- MLflow runs screenshot