MLOps Heart Disease Prediction

Group – 53

SOUMYA RANJAN MOHANTY : 2024AA05498
DHUPKAR SAURABH SHARACHCHANDRA : 2024AA05496
SRIDHAR R : 2024AA05357
TANMAY RASTOGI : 2024AA05742
SUMANGALA B S : 2023AC05941

1. Setup & Installation Instructions :

a) Ensure the following are installed on the system:

- Mini conda (or any other virtual environment system)
- Python **3.10+ (Installed within environment cmd : conda create -n flops python -y , conda activate mlops)**
- Git
- Docker Desktop (with Kubernetes enabled)
- kubectl

b) pip install -r requirements.txt within the environment “mlops”

2. Exploratory Data Analysis (EDA) & Modelling Choices :

a) Exploratory Data Analysis (EDA):

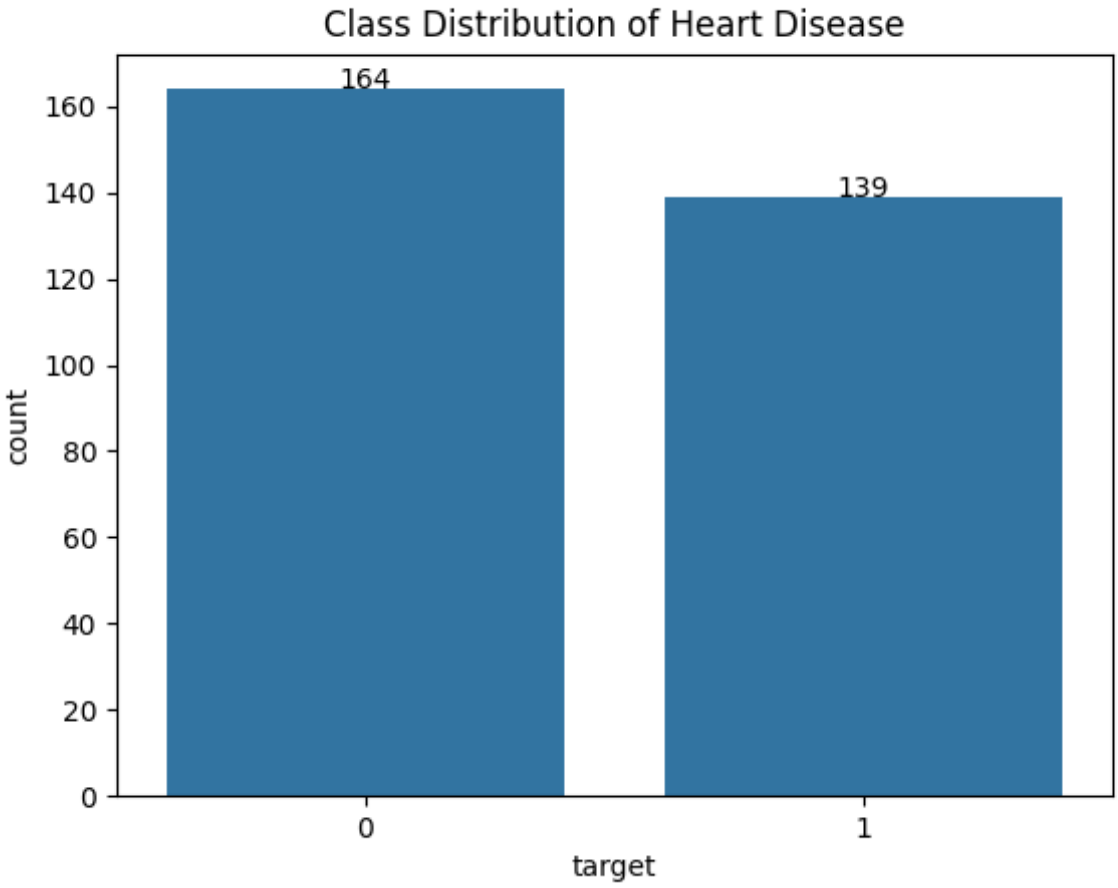
The initial exploratory data analysis was performed to understand the structure, distribution, and quality of the dataset.

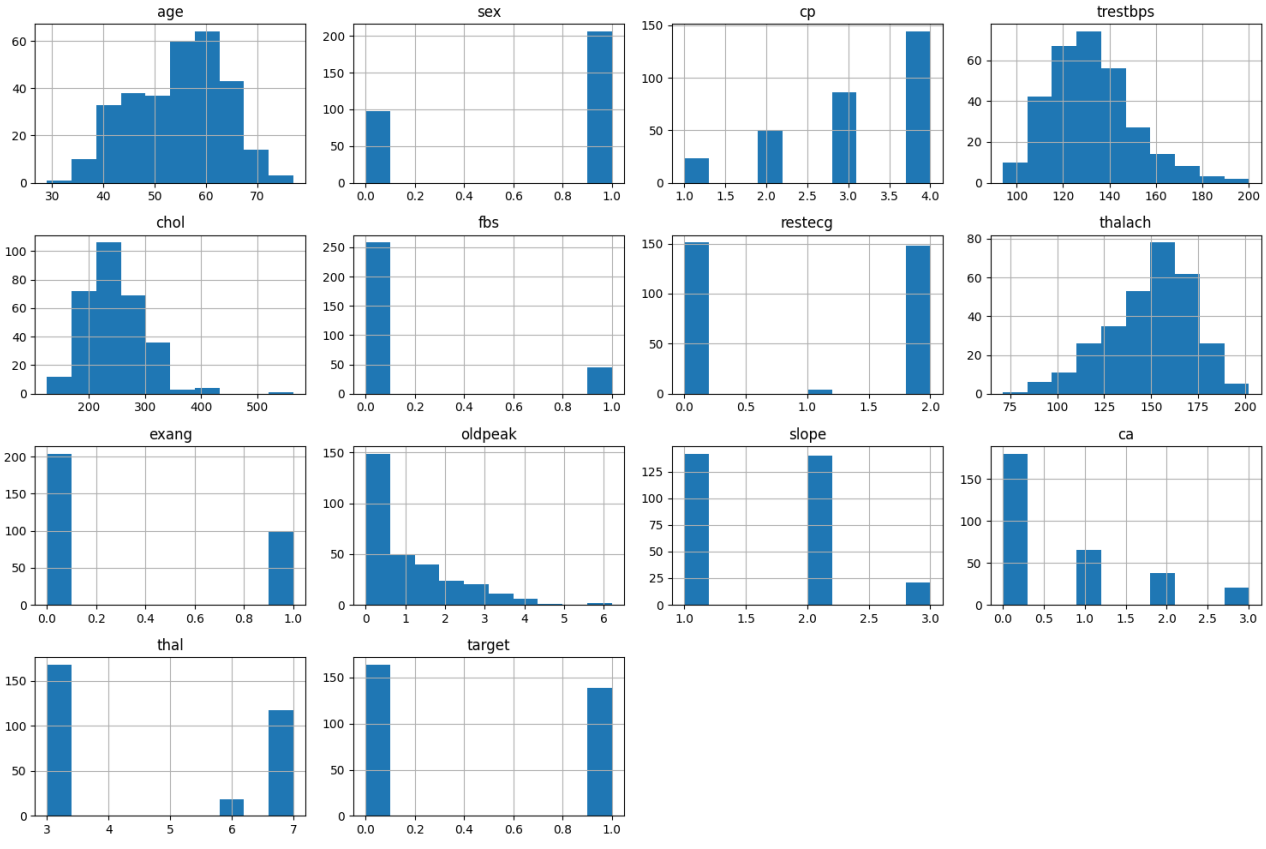
Key EDA observations:

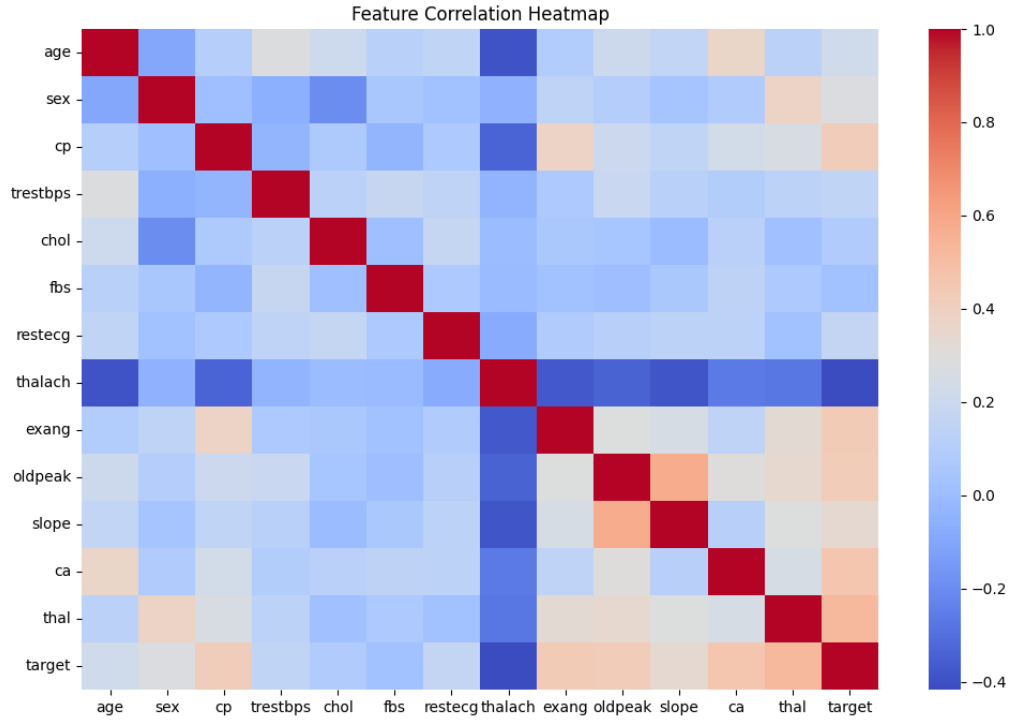
- The dataset contains clinical and demographic attributes related to heart disease.
- The target variable (target) is binary, representing the presence or absence of heart disease.
- Numerical features such as age, chol, trestbps, thalach, and oldpeak exhibited varying scales and distributions.
- Categorical and ordinal features (e.g., cp, restecg, slope, thal) were already encoded numerically but represented discrete clinical categories.
- No missing values were present in the processed dataset, ensuring suitability for direct model training.

EDA conclusions:

- Feature scaling was required for continuous variables.
- Categorical and ordinal features could be used directly without one-hot encoding.







b) Modelling Choices

Two classification models were selected to balance interpretability and predictive performance:

1. Logistic Regression

- Chosen as a **baseline model** due to its simplicity and interpretability.
- Commonly used in medical and clinical prediction tasks.
- Provides a linear decision boundary and probabilistic outputs.
- Helps establish a benchmark for more complex models.

2. Random Forest Classifier

- Selected as a **non-linear ensemble model** capable of capturing complex feature interactions.
- Robust to feature scaling and noise.
- Provides improved predictive performance compared to linear models.

3. Feature Engineering & Preprocessing

- Preprocessing was implemented using a scikit-learn Pipeline combined with a ColumnTransformer.
- Continuous features were standardized using StandardScaler.
- Binary, ordinal, and discrete numerical features were passed through without transformation.
- This ensured consistent preprocessing during both training and inference.

4. Model Selection Strategy

- Models were evaluated using **cross-validation**.
- Multiple metrics were considered:
    - Accuracy
    - Precision
    - Recall
    - ROC-AUC
- A **multi-metric selection strategy** was applied, prioritizing ROC-AUC recall to account for the clinical importance of minimizing false negatives.

3. Experiment Tracking Using MLflow

MLflow was used to manage and track machine learning experiments throughout the model development process. This enabled systematic comparison of different models and ensured reproducibility of results.

a) Tracked Experiments :

Two independent experiments were conducted:

1. **Logistic Regression**
2. **Random Forest Classifier**

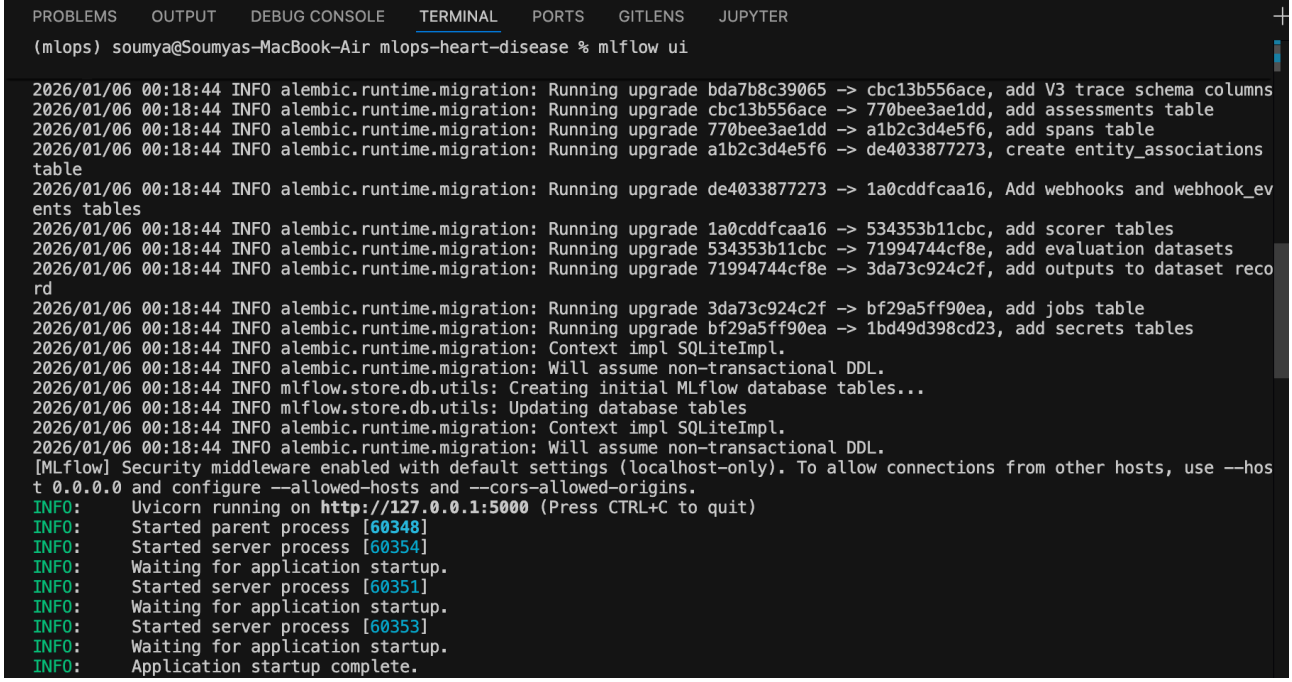
Each model training run was logged as a **separate MLflow run** under a common experiment.

b) Parameters Logged :

For each run, the following parameters were logged:

- Model type (e.g., Logistic Regression, Random Forest)
- Model-specific hyperparameters (e.g., max_iter, n_estimators)

c) Metrics Logged :



Each experiment recorded the following evaluation metrics using cross-validation:

- Accuracy
- Precision
- Recall
- ROC-AUC

These metrics allowed objective comparison between models based on both predictive performance and clinical relevance.

d) Artifacts Logged

MLflow automatically stored:

- The trained model pipeline (including preprocessing steps)
- Environment metadata required to reload the model

This ensured that each experiment could be reproduced exactly at a later time.

e) Model Comparison & Selection

- MLflow's UI was used to compare runs visually.
- A multi-metric evaluation strategy was applied to select the best-performing model.
- The final selected model was serialized separately as a production-ready artifact (final_model.pkl).

1. Prefer higher ROC-AUC
2. If close (±0.02), choose higher Recall
3. If still close, choose higher Precision

f) Benefits of Using MLflow

- Centralized experiment tracking
- Clear separation of training runs
- Reproducibility through logged artifacts
- Transparent comparison of model performance

g) Reproducibility

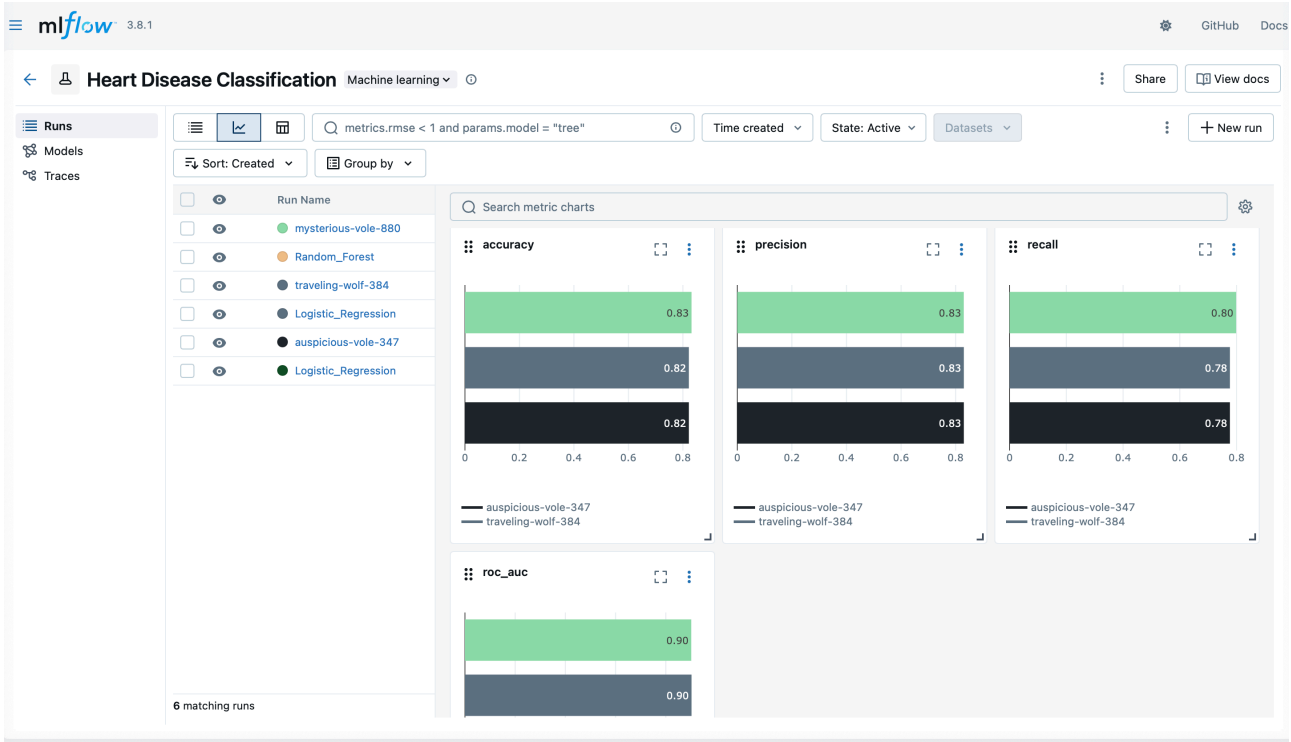
Because preprocessing, training, and evaluation were encapsulated in a single pipeline and logged using MLflow, the experiments can be reliably reproduced across environments.

3. Architecture Diagram:

The architecture follows a modular MLOps design. Raw data is first processed using a reproducible preprocessing pipeline implemented with scikit-learn.

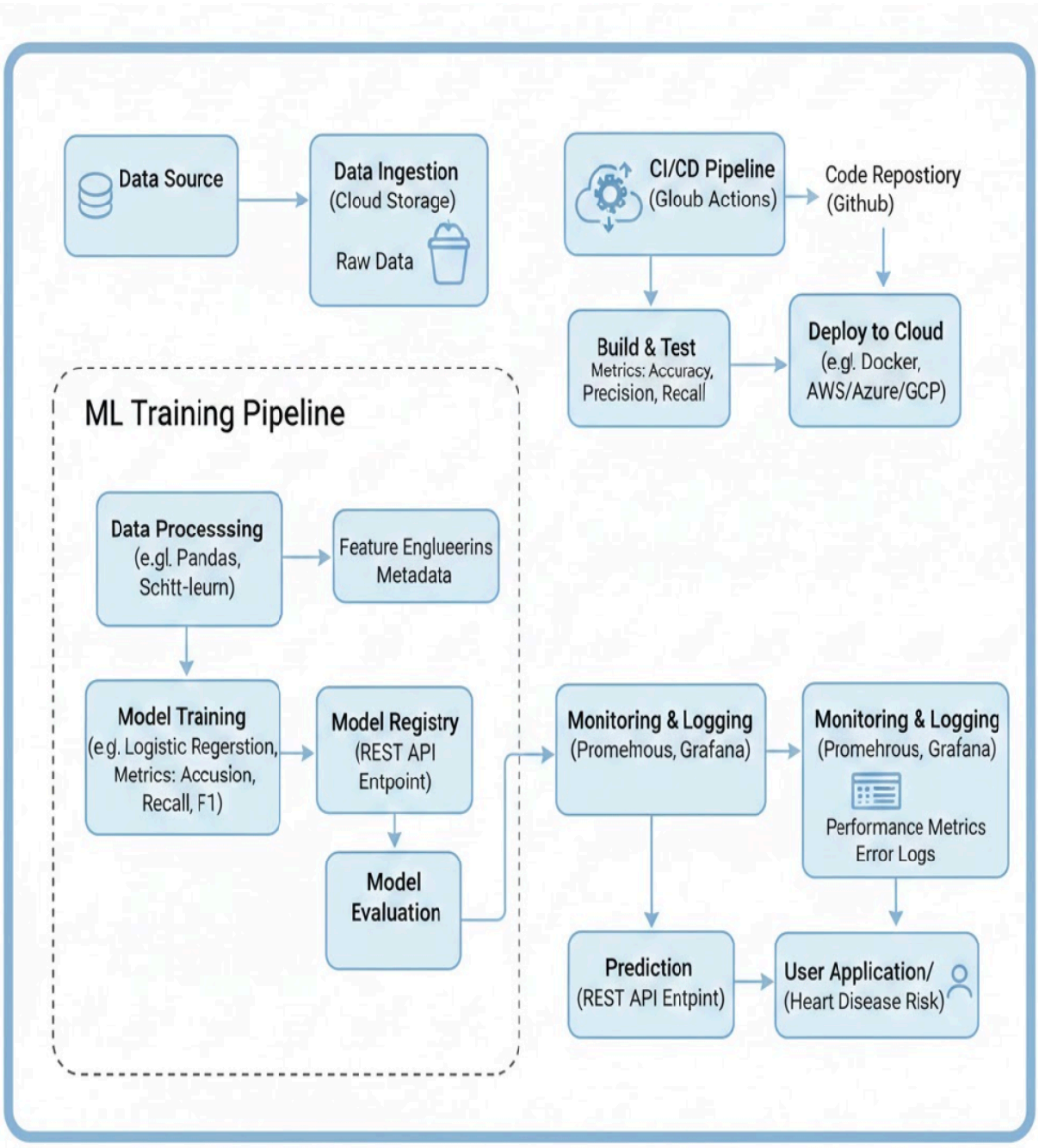
Multiple machine learning models are trained and tracked using MLflow, which logs parameters, metrics, and artifacts. A multi-metric evaluation strategy selects the best performing model, which is serialized as a standalone artifact.

A CI/CD pipeline implemented using GitHub Actions automates testing, training, and artifact generation.

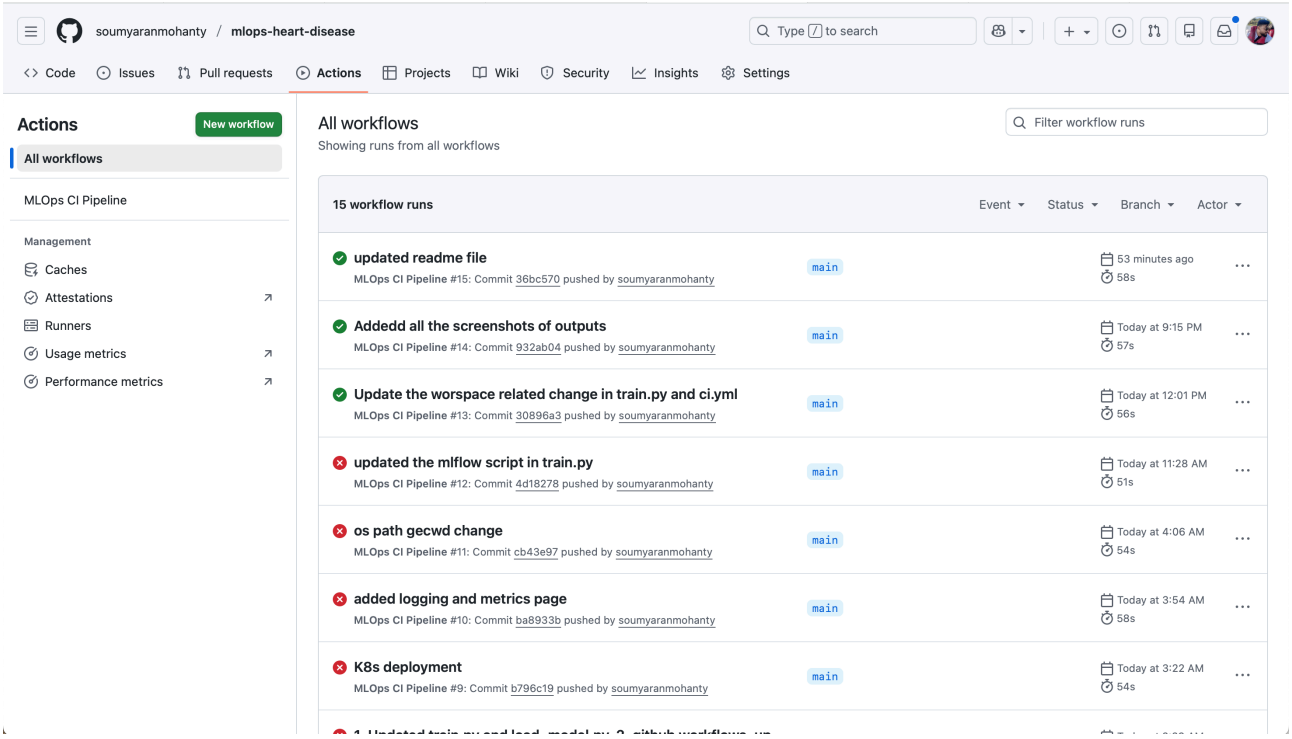


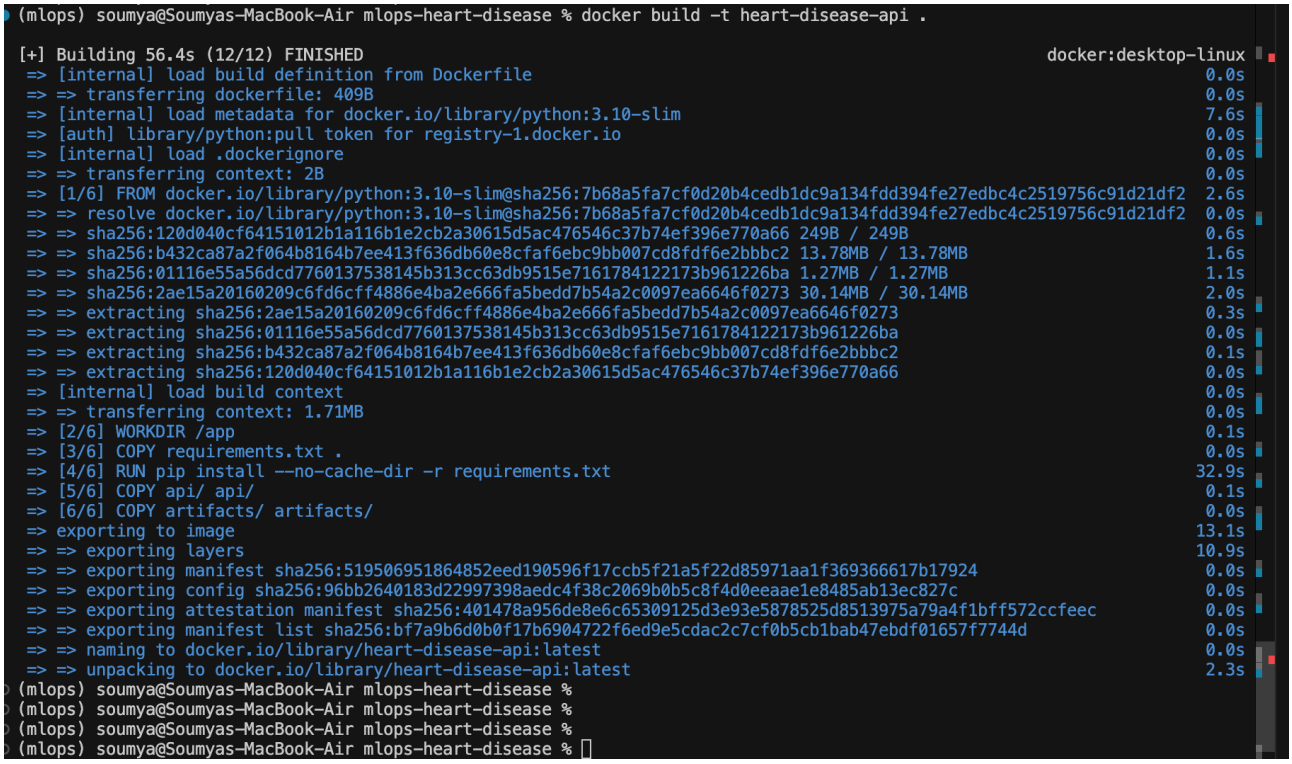
The final model is served using a FastAPI application, containerized with Docker, and deployed to a Kubernetes cluster.

Monitoring is enabled through application logs, a metrics endpoint, and optional Prometheus integration, providing observability into runtime behavior.



5. CI/CD and deployment workflow screenshots.





```
(mlops) soumya@Soumyas-MacBook-Air mlops-heart-disease % kubectl config current-context

docker-desktop
(mlops) soumya@Soumyas-MacBook-Air mlops-heart-disease %
(mlops) soumya@Soumyas-MacBook-Air mlops-heart-disease %
(mlops) soumya@Soumyas-MacBook-Air mlops-heart-disease % kubectl get services
NAME                     TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
heart-disease-service    NodePort    10.111.72.156   <none>        80:30007/TCP   3m49s
kubernetes               ClusterIP   10.96.0.1       <none>        443/TCP        16m
(mlops) soumya@Soumyas-MacBook-Air mlops-heart-disease %



deployment.apps/heart-disease-api created
(mlops) soumya@Soumyas-MacBook-Air mlops-heart-disease %
(mlops) soumya@Soumyas-MacBook-Air mlops-heart-disease % kubectl get deployments
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
heart-disease-api   1/1     1            1           18s
(mlops) soumya@Soumyas-MacBook-Air mlops-heart-disease %
(mlops) soumya@Soumyas-MacBook-Air mlops-heart-disease %
(mlops) soumya@Soumyas-MacBook-Air mlops-heart-disease % kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
heart-disease-api-67c45ddd78-qndtv   1/1     Running   0          26s
(mlops) soumya@Soumyas-MacBook-Air mlops-heart-disease %
(mlops) soumya@Soumyas-MacBook-Air mlops-heart-disease %
```

```
(mlops) soumya@Soumyas-MacBook-Air mlops-heart-disease % kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
heart-disease-api-67c45ddd78-qndtv   1/1     Running   0          12m
(mlops) soumya@Soumyas-MacBook-Air mlops-heart-disease %
```

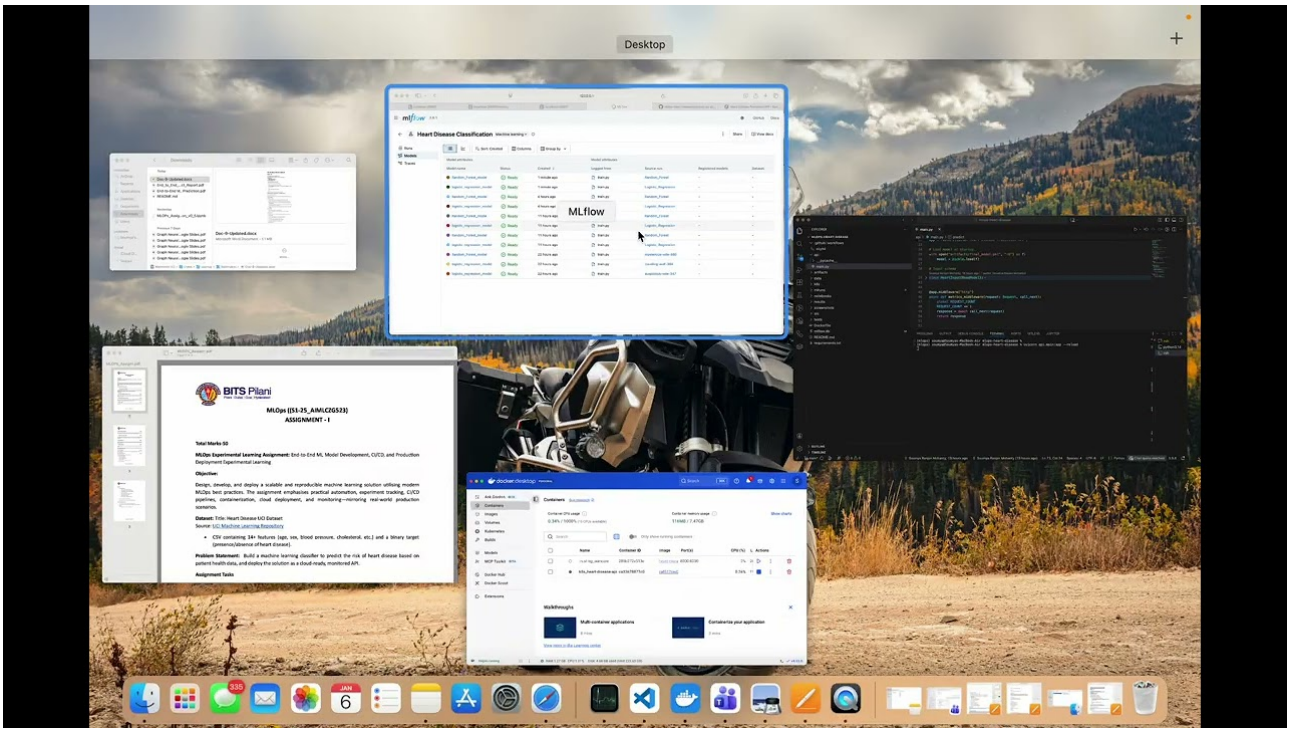
```
(mlops) soumya@Soumyas-MacBook-Air mlops-heart-disease % kubectl get nodes -o wide

NAME             STATUS   ROLES           AGE   VERSION   INTERNAL-IP    EXTERNAL-IP   OS-IMAGE         KERNEL-VERSION     CO
NTAINER-RUNTIME
docker-desktop   Ready    control-plane   13m   v1.34.1   192.168.65.3   <none>        Docker Desktop   6.12.54-linuxkit   do
cker://29.1.3
(mlops) soumya@Soumyas-MacBook-Air mlops-heart-disease %
(mlops) soumya@Soumyas-MacBook-Air mlops-heart-disease %
(mlops) soumya@Soumyas-MacBook-Air mlops-heart-disease %
(mlops) soumya@Soumyas-MacBook-Air mlops-heart-disease % curl -X POST "http://localhost:30007/predict" \
-H "Content-Type: application/json" \
-d '{
  "age": 55,
  "sex": 1,
  "cp": 2,
  "trestbps": 130,
  "chol": 250,
  "fbs": 0,
  "restecg": 1,
  "thalach": 150,
  "exang": 0,
  "oldpeak": 1.5,
  "slope": 2,
  "ca": 0,
  "thal": 3
}'
{"prediction":0,"probability":0.15}%
(mlops) soumya@Soumyas-MacBook-Air mlops-heart-disease %
(mlops) soumya@Soumyas-MacBook-Air mlops-heart-disease %
(mlops) soumya@Soumyas-MacBook-Air mlops-heart-disease %
```

6. Link to code repository :

https://github.com/soumyaranmohanty/mlops-heart-disease

7. Video Link :


Desktop
MLflow
BITS Pilani
MLOps (S1-25_AIMLCZG523)
ASSIGNMENT - 1