❤️ Heart Disease Prediction — ML Classification

Machine Learning — Assignment 2

M.Tech (AIML) — BITS Pilani, Work Integrated Learning Programmes

Student ID: 2025AB05189

Submission Date: 15-Feb-2026

1. GitHub Repository Link

📁 **GitHub Repository:**
https://github.com/2025ab05189-gokul/2025ab05189-ml-assignment-2

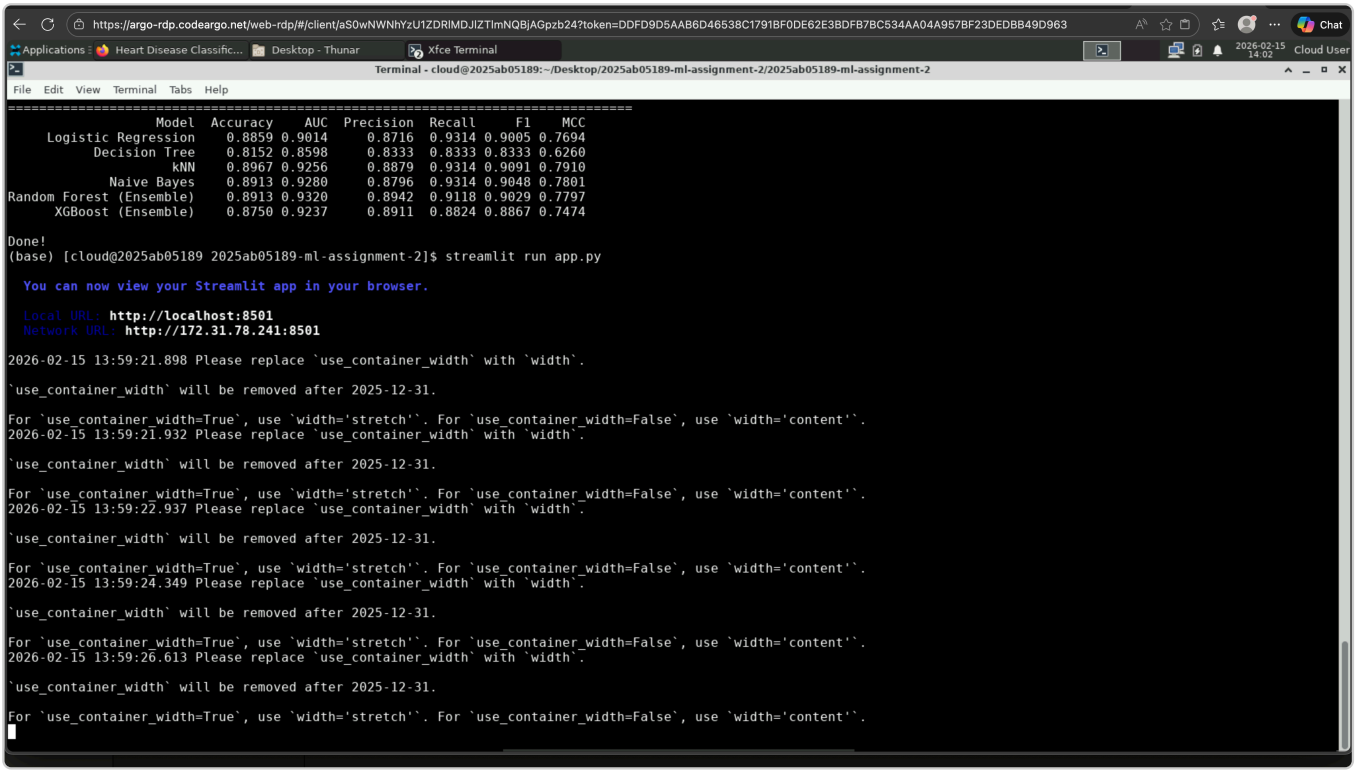
Contains: app.py, requirements.txt, README.md, model/model_training.py, data/heart.csv

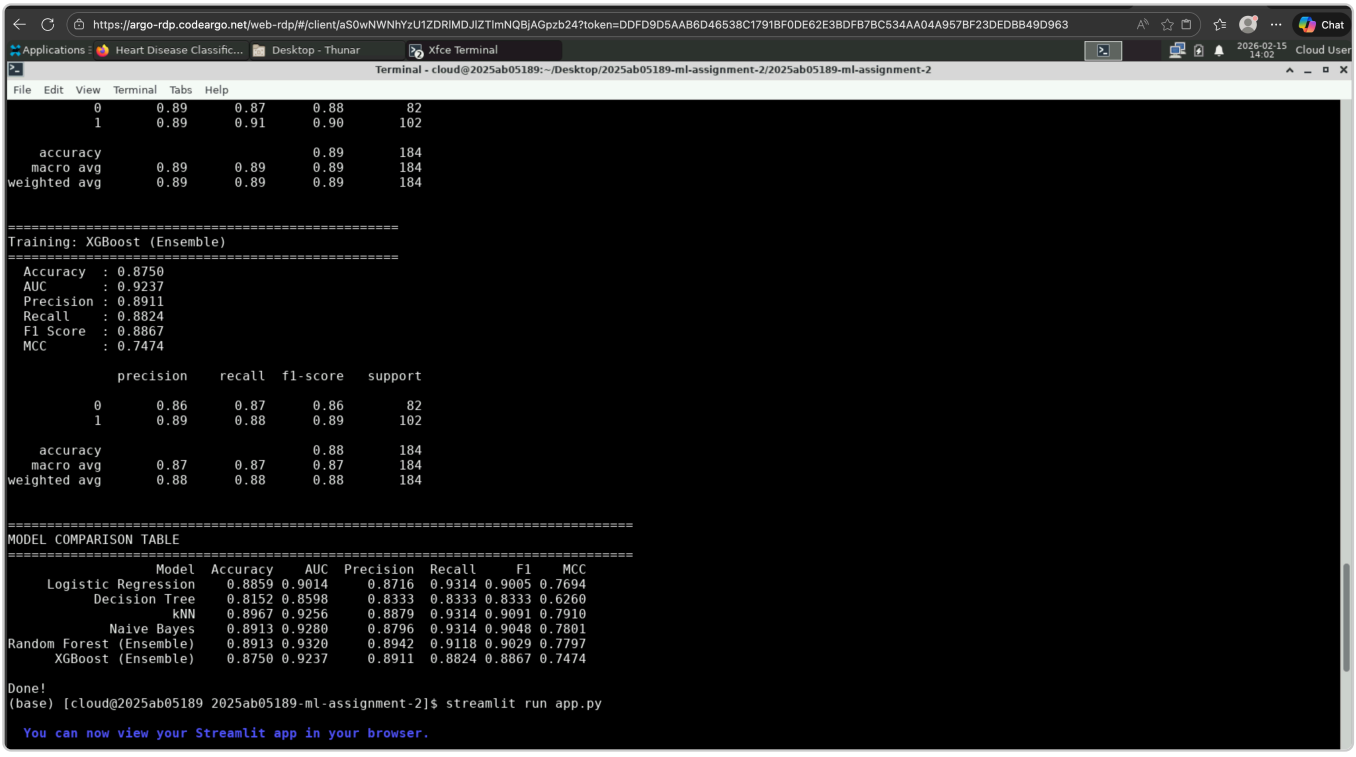
2. Live Streamlit App Link

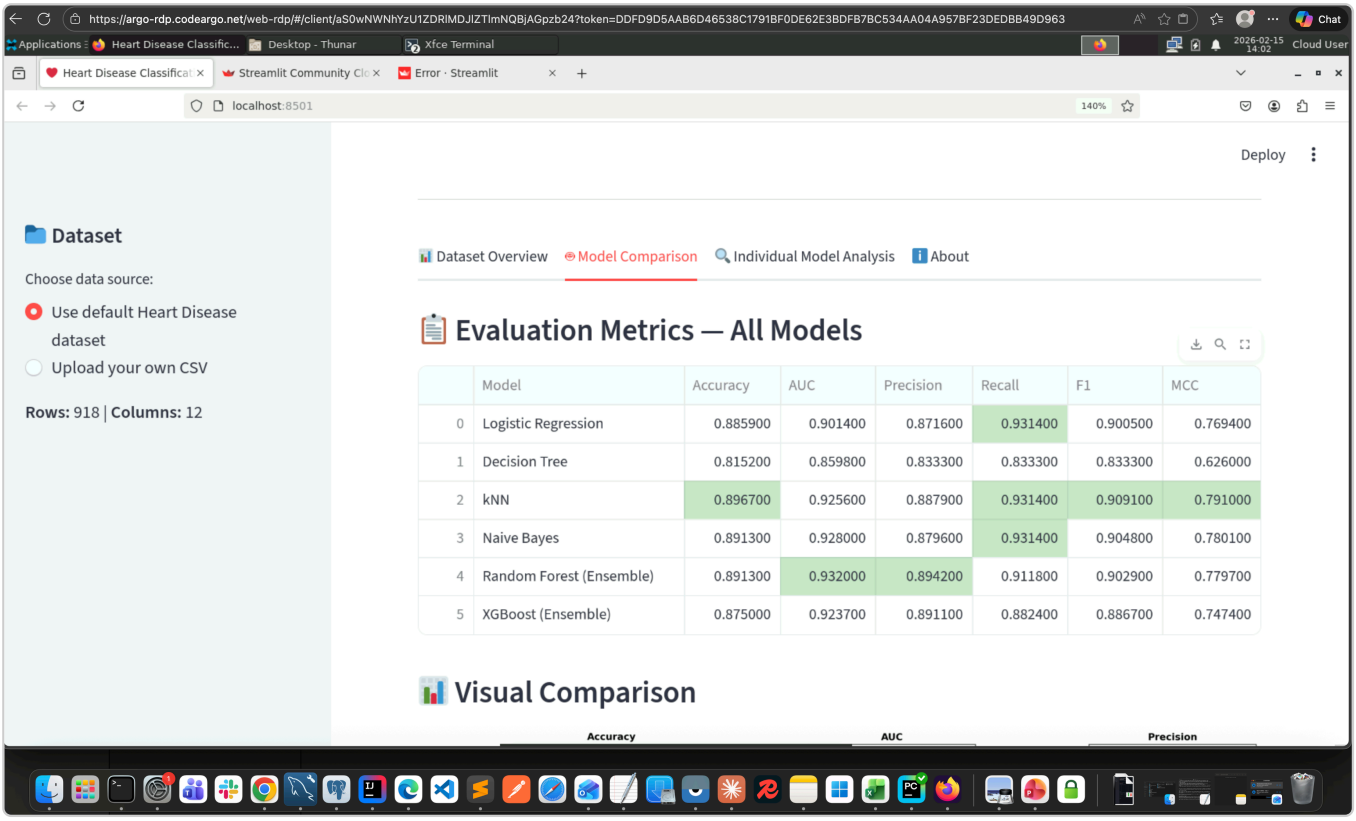
🌐 **Live Streamlit App:**
https://2025ab05189-ml-assignment-2-2nwpjziuprlqztegu6m4t9.streamlit.app/

Deployed on Streamlit Community Cloud — opens interactive frontend when clicked

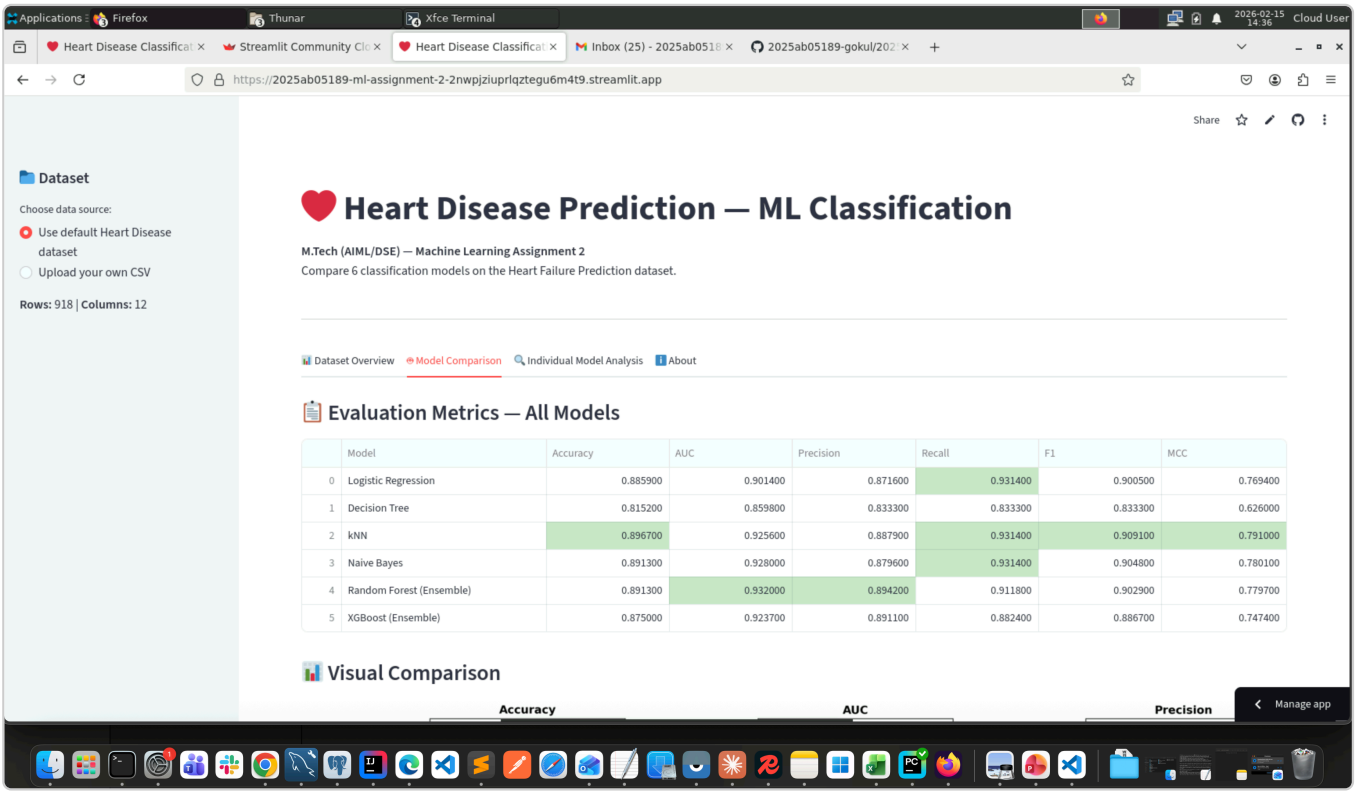
3. BITS Virtual Lab — Execution Screenshots


https://argo-rdp.codeargo.net/web-rdp/#/client/aS0wNWNhYzU1ZDRlMDJlZTlmNQBjAGpzb24?token=DDFD9D5AAB6D46538C1791BF0DE62E3BDFB7BC534AA04A957BF23DEDBB49D963
Terminal - cloud@2025ab05189:~/Desktop/2025ab05189-ml-assignment-2/2025ab05189-ml-assignment-2
File Edit View Terminal Tabs Help
Model Accuracy AUC Precision Recall F1 MCC
Logistic Regression 0.8859 0.9014 0.8716 0.9314 0.9005 0.7694
Decision Tree 0.8152 0.8598 0.8333 0.8333 0.8333 0.6260
kNN 0.8967 0.9256 0.8879 0.9314 0.9091 0.7910
Naive Bayes 0.8913 0.9280 0.8796 0.9314 0.9048 0.7801
Random Forest (Ensemble) 0.8913 0.9320 0.8942 0.9118 0.9029 0.7797
XGBoost (Ensemble) 0.8750 0.9237 0.8911 0.8824 0.8867 0.7474
Done!
(base) [cloud@2025ab05189 2025ab05189-ml-assignment-2]$ streamlit run app.py
You can now view your Streamlit app in your browser.
Local URL: http://localhost:8501
Network URL: http://172.31.78.241:8501
2026-02-15 13:59:21.898 Please replace `use_container_width` with `width`.
`use_container_width` will be removed after 2025-12-31.
For `use_container_width=True`, use `width='stretch'`. For `use_container_width=False`, use `width='content'`.
2026-02-15 13:59:21.932 Please replace `use_container_width` with `width`.
`use_container_width` will be removed after 2025-12-31.
For `use_container_width=True`, use `width='stretch'`. For `use_container_width=False`, use `width='content'`.
2026-02-15 13:59:22.937 Please replace `use_container_width` with `width`.
`use_container_width` will be removed after 2025-12-31.
For `use_container_width=True`, use `width='stretch'`. For `use_container_width=False`, use `width='content'`.
2026-02-15 13:59:24.349 Please replace `use_container_width` with `width`.
`use_container_width` will be removed after 2025-12-31.
For `use_container_width=True`, use `width='stretch'`. For `use_container_width=False`, use `width='content'`.
2026-02-15 13:59:26.613 Please replace `use_container_width` with `width`.
`use_container_width` will be removed after 2025-12-31.
For `use_container_width=True`, use `width='stretch'`. For `use_container_width=False`, use `width='content'`.


https://argo-rdp.codeargo.net/web-rdp/#/client/aS0wNWNhYzU1ZDRlMDJlZTlmNQBjAGpzb24?token=DDFD9D5AAB6D46538C1791BF0DE62E3BDFB7BC534AA04A957BF23DEDBB49D963
Terminal - cloud@2025ab05189:~/Desktop/2025ab05189-ml-assignment-2/2025ab05189-ml-assignment-2
File Edit View Terminal Tabs Help
0 0.89 0.87 0.88 82
1 0.89 0.91 0.90 102
accuracy 0.89 184
macro avg 0.89 0.89 0.89 184
weighted avg 0.89 0.89 0.89 184
Training: XGBoost (Ensemble)
Accuracy : 0.8750
AUC : 0.9237
Precision : 0.8911
Recall : 0.8824
F1 Score : 0.8867
MCC : 0.7474
precision recall f1-score support
0 0.86 0.87 0.86 82
1 0.89 0.88 0.89 102
accuracy 0.88 184
macro avg 0.87 0.87 0.87 184
weighted avg 0.88 0.88 0.88 184
MODEL COMPARISON TABLE
Model Accuracy AUC Precision Recall F1 MCC
Logistic Regression 0.8859 0.9014 0.8716 0.9314 0.9005 0.7694
Decision Tree 0.8152 0.8598 0.8333 0.8333 0.8333 0.6260
kNN 0.8967 0.9256 0.8879 0.9314 0.9091 0.7910
Naive Bayes 0.8913 0.9280 0.8796 0.9314 0.9048 0.7801
Random Forest (Ensemble) 0.8913 0.9320 0.8942 0.9118 0.9029 0.7797
XGBoost (Ensemble) 0.8750 0.9237 0.8911 0.8824 0.8867 0.7474
Done!
(base) [cloud@2025ab05189 2025ab05189-ml-assignment-2]$ streamlit run app.py
You can now view your Streamlit app in your browser.



	Model	Accuracy	AUC	Precision	Recall	F1	MCC
0	Logistic Regression	0.885900	0.901400	0.871600	0.931400	0.900500	0.769400
1	Decision Tree	0.815200	0.859800	0.833300	0.833300	0.833300	0.626000
2	kNN	0.896700	0.925600	0.887900	0.931400	0.909100	0.791000
3	Naive Bayes	0.891300	0.928000	0.879600	0.931400	0.904800	0.780100
4	Random Forest (Ensemble)	0.891300	0.932000	0.894200	0.911800	0.902900	0.779700
5	XGBoost (Ensemble)	0.875000	0.923700	0.891100	0.882400	0.886700	0.747400



	Model	Accuracy	AUC	Precision	Recall	F1	MCC
0	Logistic Regression	0.885900	0.901400	0.871600	0.931400	0.900500	0.769400
1	Decision Tree	0.815200	0.859800	0.833300	0.833300	0.833300	0.626000
2	kNN	0.896700	0.925600	0.887900	0.931400	0.909100	0.791000
3	Naive Bayes	0.891300	0.928000	0.879600	0.931400	0.904800	0.780100
4	Random Forest (Ensemble)	0.891300	0.932000	0.894200	0.911800	0.902900	0.779700
5	XGBoost (Ensemble)	0.875000	0.923700	0.891100	0.882400	0.886700	0.747400

4. README.md Content

a. Problem Statement

Heart disease is one of the leading causes of death globally. Early detection using clinical parameters can save lives. The goal of this project is to build and compare multiple machine learning classification models that predict whether a patient has heart disease based on 11 clinical features.

We implement 6 different classifiers, evaluate them using 6 standard metrics, and deploy an interactive Streamlit web application for demonstration.

b. Dataset Description

Property	Details
Name	Heart Failure Prediction Dataset
Source	Kaggle — fedesoriano (https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction)
Origin	Combination of 5 independent heart disease datasets (Cleveland, Hungarian, Switzerland, Long Beach VA, Stalog)
Instances	918 (after removing duplicates from 1190 combined records)
Features	11 clinical features + 1 binary target
Target	`target` — 1 (heart disease, 508 cases) / 0 (no heart disease, 410 cases)
Task	Binary Classification
Train/Test Split	80/20 stratified (734 train, 184 test)

Feature Details

Feature	Description	Type
`age`	Age of the patient (years)	Numerical
`sex`	Sex of the patient (1=Male, 0=Female)	Binary
`chest_pain_type`	Chest pain type (1: Typical Angina, 2: Atypical Angina, 3: Non-Anginal Pain, 4: Asymptomatic)	Categorical
`resting_bp`	Resting blood pressure (mm Hg)	Numerical

`cholesterol`	Serum cholesterol (mg/dl)	Numerical
`fasting_bs`	Fasting blood sugar > 120 mg/dl (1=True, 0=False)	Binary
`resting_ecg`	Resting ECG results (0: Normal, 1: ST-T abnormality, 2: LVH)	Categorical
`max_hr`	Maximum heart rate achieved	Numerical
`exercise_angina`	Exercise-induced angina (1=Yes, 0=No)	Binary
`oldpeak`	ST depression induced by exercise	Numerical
`st_slope`	Slope of peak exercise ST segment (1: Up, 2: Flat, 3: Down)	Categorical

c. Models Used

Comparison Table — Evaluation Metrics

ML Model Name	Accuracy	AUC	Precision	Recall	F1	MCC
Logistic Regression	0.8859	0.9014	0.8716	0.9314	0.9005	0.7694
Decision Tree	0.8152	0.8598	0.8333	0.8333	0.8333	0.6260
kNN	0.8967	0.9256	0.8879	0.9314	0.9091	0.7910
Naive Bayes	0.8913	0.9280	0.8796	0.9314	0.9048	0.7801
Random Forest (Ensemble)	0.8913	0.9320	0.8942	0.9118	0.9029	0.7797
XGBoost (Ensemble)	0.8750	0.9237	0.8911	0.8824	0.8867	0.7474

Model Performance Observations

ML Model Name	Observation
Logistic Regression	Achieved 88.59% accuracy and serves as a strong interpretable baseline. High recall (0.9314) means it correctly identifies 93% of heart disease patients. AUC of 0.9014 confirms good discriminative ability. The linear decision boundary generalizes well on this dataset, making it suitable for clinical deployment where model transparency is required.

Decision Tree	Lowest accuracy (81.52%) and AUC (0.8598) among all models, indicating high variance and overfitting tendencies even with max_depth=5. Balanced precision and recall (both 0.8333) but the lowest MCC (0.6260) reflects weaker overall classification quality. Most valuable for interpretability via tree visualization rather than raw predictive performance.
kNN	Achieved the highest accuracy (89.67%) among all models. Achieved the same high recall (0.9314) as Logistic Regression and Naive Bayes. AUC of 0.9256 is strong. Performance is heavily dependent on feature scaling (StandardScaler applied) and the choice of k=7. The instance-based approach captures local data patterns effectively for this dataset size.
Naive Bayes	Achieved 89.13% accuracy with the joint-highest recall (0.9314) — catching 93% of disease cases. AUC of 0.9280 is the second-highest, indicating well-calibrated probability estimates. Fast training time makes it practical for real-time applications. The feature independence assumption slightly limits precision compared to ensemble methods.
Random Forest (Ensemble)	Achieved 89.13% accuracy and the highest AUC (0.9320) among all models. Best precision (0.8942) indicates fewer false positives. Bagging 200 decorrelated trees significantly reduces the variance problem seen in the single Decision Tree (+7.61% accuracy improvement). MCC of 0.7797 reflects strong balanced performance across both classes.
XGBoost (Ensemble)	Achieved 87.50% accuracy with the highest precision (0.8911) but the lowest recall (0.8824) among the top models. AUC of 0.9237 is competitive. The gradient boosting approach with L1/L2 regularization provides good generalization. Lower recall compared to other models suggests a more conservative decision threshold; tuning the threshold could improve sensitivity for clinical use.

Project Structure

```
2025ab05189-ml-assignment-2/
├── app.py                    # Streamlit web application
├── requirements.txt          # Python dependencies
├── README.md                 # Project documentation
├── data/
│   └── heart.csv             # Local copy of dataset (fallback)
├── model/
│   └── model_training.py     # Training script for all 6 models
```

Streamlit App Features

Feature	Description
Dataset upload (CSV)	Upload test data or use the default dataset
Model selection dropdown	Choose any of the 6 implemented models for detailed analysis
Evaluation metrics display	Accuracy, AUC, Precision, Recall, F1, MCC for all models
Confusion matrix & classification report	Per-model detailed analysis with heatmap visualization
ROC curve comparison	Overlay of all model ROC curves
Visual bar chart comparison	Side-by-side metric comparison across all 6 models

Deployment

Deployed on **Streamlit Community Cloud**: https://2025ab05189-ml-assignment-2-2nwpjziuprlqztegu6m4t9.streamlit.app/

Tech Stack

Python 3.10+, Streamlit, scikit-learn, XGBoost, pandas, numpy, matplotlib, seaborn