Heart Disease Prediction Report

End-to-End Machine Learning Project

Team Members: Abdelrhman Hammoudeh & Yasser Abushaikha
Team University numbers: 0248546 & 0230067

Course: AI and Machine Learning
University of Jordan
Date: December 04, 2025
Report compiled using Typst

Project Repository:
https://github.com/Abdelrhman-Hammoudeh/Project_AI

Source Code: GitHub

Table of Contents

Source Code: GitHub

1. 1. Introduction

Cardiovascular diseases remain the leading cause of mortality worldwide. Early detection through predictive modeling offers a scalable pathway for risk stratification and preventive intervention. This project develops an end-to-end machine learning pipeline to predict the presence or absence of heart disease using the UCI Heart Disease Dataset.

Our methodology strictly separates the **Exploratory Data Analysis (DS Phase)** from the **Machine Learning Engineering (ML Phase)**. This architectural choice prevents data leakage, ensures reproducibility, and follows industry best practices in applied machine learning.

1.1. 1.1 Project Objectives

1. Conduct rigorous exploratory analysis to understand data quality and feature relationships.
2. Formulate evidence-based preprocessing decisions grounded in data observations.
3. Implement a leak-free machine learning pipeline using Scikit-learn.
4. Evaluate model performance using multiple metrics (Accuracy, Recall, AUC).
5. Provide clinical interpretation of results and feature importance.

1.2. 1.2 Dataset & Scope

- **Dataset:** UCI Heart Disease Dataset (combined from 4 hospitals)
- **Samples:** 920 patients
- **Features:** 16 clinical attributes
- **Target:** Binary classification (Healthy vs Diseased)

2. 2. Dataset Overview

The UCI Heart Disease Dataset is a well-established benchmark in medical AI research. It aggregates patient records from four independent medical centers: Cleveland Clinic, Hungarian Institute of Cardiology, University Hospital Zurich, and VA Medical Center (Long Beach).

2.1. 2.1 Data Structure

Attribute	Type	Notes
Samples	Numeric	920 patient records
Features	Mixed	16 total: 5 numeric, 8 categorical, 3 identifiers
Missing Values	Yes	`ca` (66%), `thal` (53%), `slope` (34%)
Target Variable	Binary	`num`: 0 = Healthy, 1-4 = Diseased (binarized)
Class Balance	Imbalanced	≈45% Healthy, ≈55% Diseased

Table 1: Dataset Summary

Source Code: GitHub

2.2. 2.2 Data Source & Ethical Considerations

All data is anonymized and publicly available at UCI Machine Learning Repository. The dataset has been used in peer-reviewed publications and serves as a standard benchmark for evaluating medical prediction algorithms.

Source Code: GitHub

3. 3. Phase I: Exploratory Data Analysis (DS Phase)

The Data Science phase focuses on **understanding** the data, not transforming it. Using Pandas, NumPy, Matplotlib, and Seaborn, we generated systematic visualizations to guide downstream ML decisions.

3.1. 3.1 Missing Data Patterns

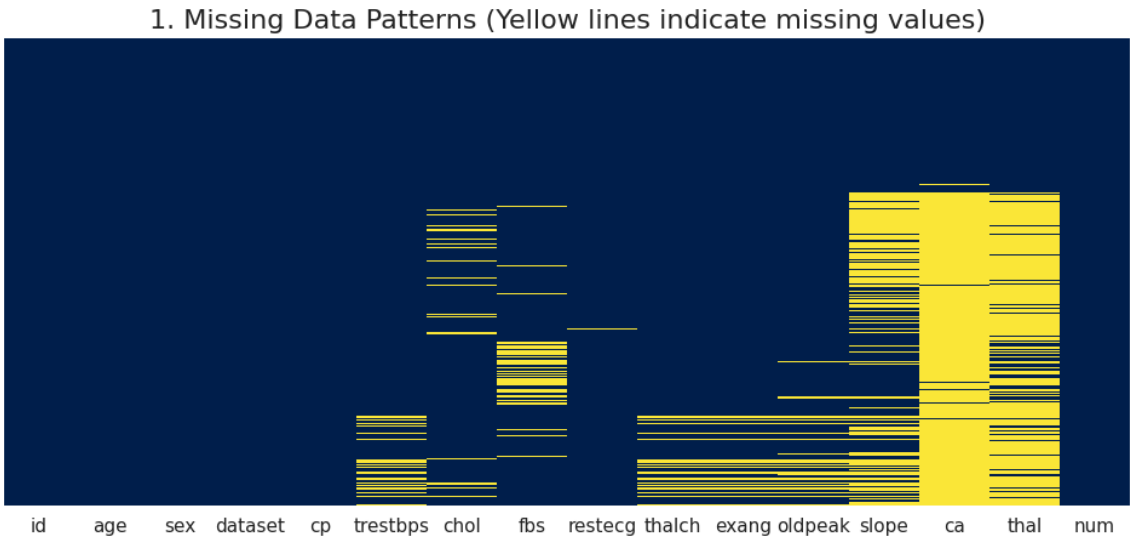

Figure 1: **Missing Data Heatmap.** Yellow horizontal lines indicate missing values. Features `ca` (coronary vessels) and `thal` (thalassemia type) exhibit 66% and 53% missingness respectively, indicating selective application of advanced diagnostics. Early-stage clinical features (`age`, `sex`, `chol`) are nearly complete.

Key Observation: The non-random pattern of missingness suggests missing values are **missing-at-random (MAR)** conditional on disease severity. Advanced diagnostic tests (angiography, thalassemia testing) are more frequently ordered for symptomatic or high-risk patients.

Implication for Preprocessing: Deletion of rows with missing values would eliminate 66% of available data. Instead, we employ sophisticated imputation.

Source Code: GitHub

3.2. 3.2 Medical Impossibilities (Sanity Check)

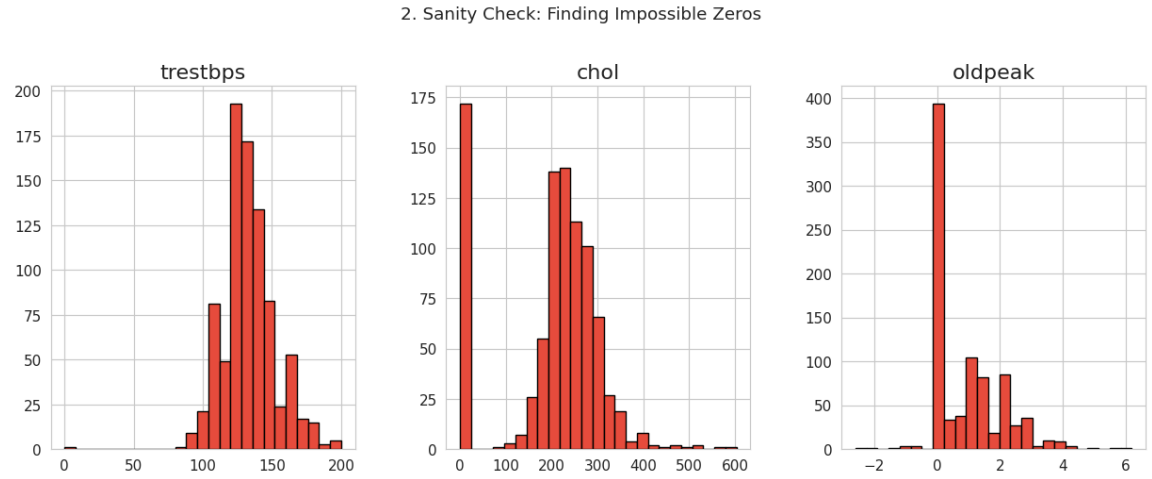

Figure 2: **Impossible Zero Values in Medical Vitals.** Histograms reveal zero occurrences in `trestbps` (resting blood pressure), `chol` (serum cholesterol), and `oldpeak` (ST-segment depression). A living patient cannot have zero blood pressure or cholesterol. These are data entry errors or encoding artifacts.

Critical Finding:

- `trestbps`: 0 appears in 5 records (likely missing value encoding)
- `chol`: 0 appears in 10 records (biologically impossible)
- `oldpeak`: Negative values (e.g., −2.6) and zeros present

Decision: Treat all such values as missing (NaN) before imputation, preventing the model from learning that "0" represents a valid low measurement.

Source Code: GitHub

3.3. 3.3 Feature Distributions

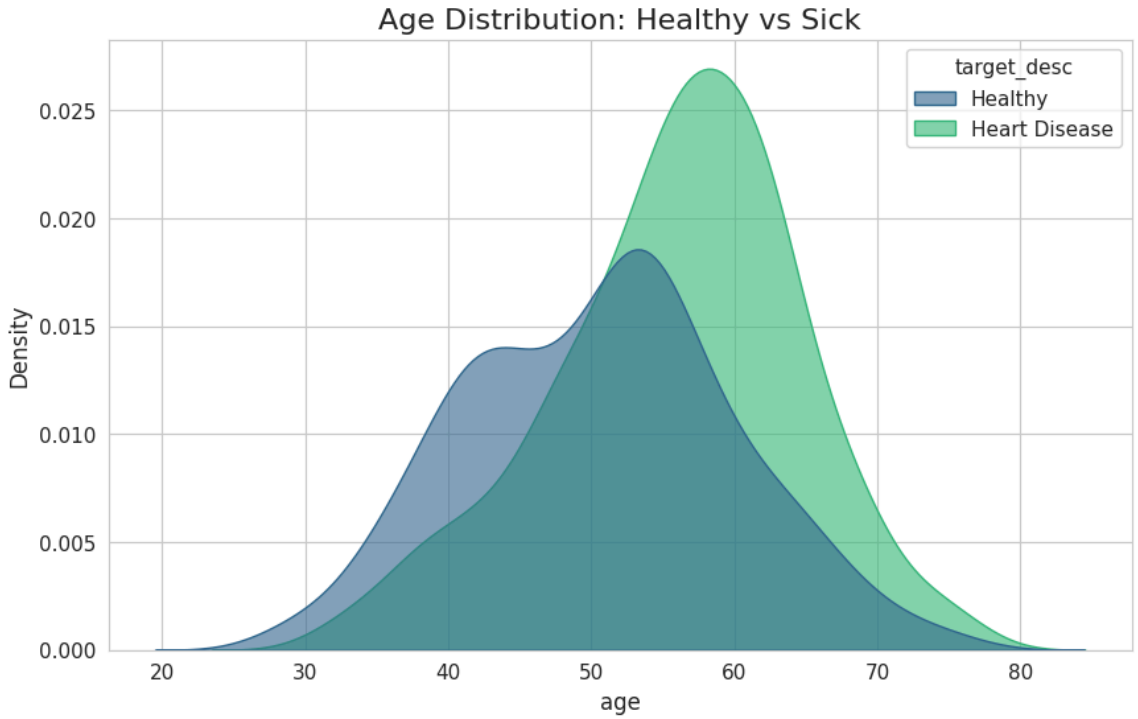

Figure 3: **Age Distribution by Disease Status.** Kernel Density Estimation (KDE) shows diseased patients (green) cluster at older ages (median ≈56 years) compared to healthy patients (blue, median ≈52 years). This ≈4-year age shift suggests age is a strong disease predictor.

Statistical Insight: The clear separation in age distributions indicates `age` will likely rank among the top features in any predictive model.

Figure 4: **Cholesterol Boxplot Analysis.** Outliers extend to 600+ mg/dL, representing rare metabolic conditions or measurement errors. These extreme values would bias mean imputation; median is more robust.

Source Code: GitHub

Preprocessing Implication: Use **Median** imputation for numerical features, not Mean. The median is resistant to outliers.

3.4. 3.4 Categorical Feature Analysis

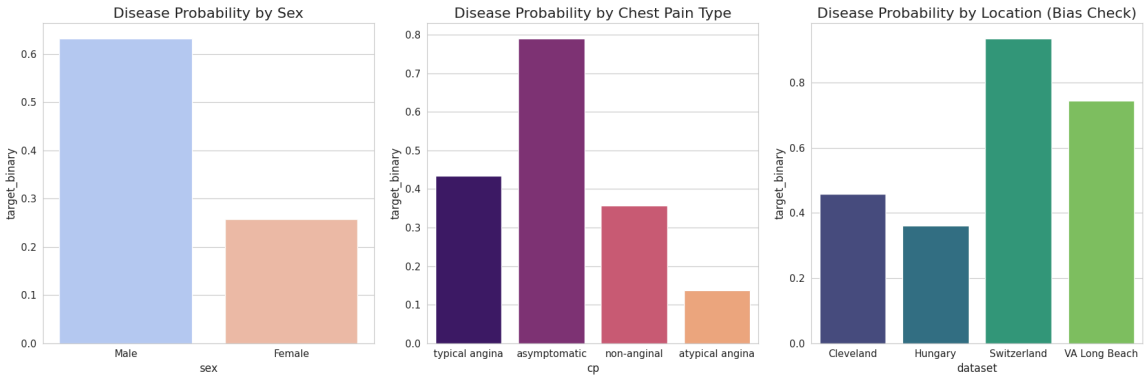

Figure 5: **(Left)** Males show 64% disease rate vs. females 26%. **(Middle)** Asymptomatic chest pain paradoxically shows highest disease rate (79%), consistent with cardiology: silent myocardial infarctions are life-threatening. **(Right)** Geographic variance in disease rates: Switzerland 91%, Cleveland 45%. This 46-point spread indicates hospital-specific bias.

Critical Observation: The `dataset` column encodes geographic/hospital-specific factors (diagnostic thresholds, patient demographics, referral patterns) rather than pure biological differences. A model trained on this feature would learn to predict **hospital identity**, not **disease presence**.

Decision: Drop the `dataset` column to prevent overfitting to spurious location patterns and ensure generalization to unseen hospitals.

Source Code: GitHub

3.5. 3.5 Feature Correlation Analysis

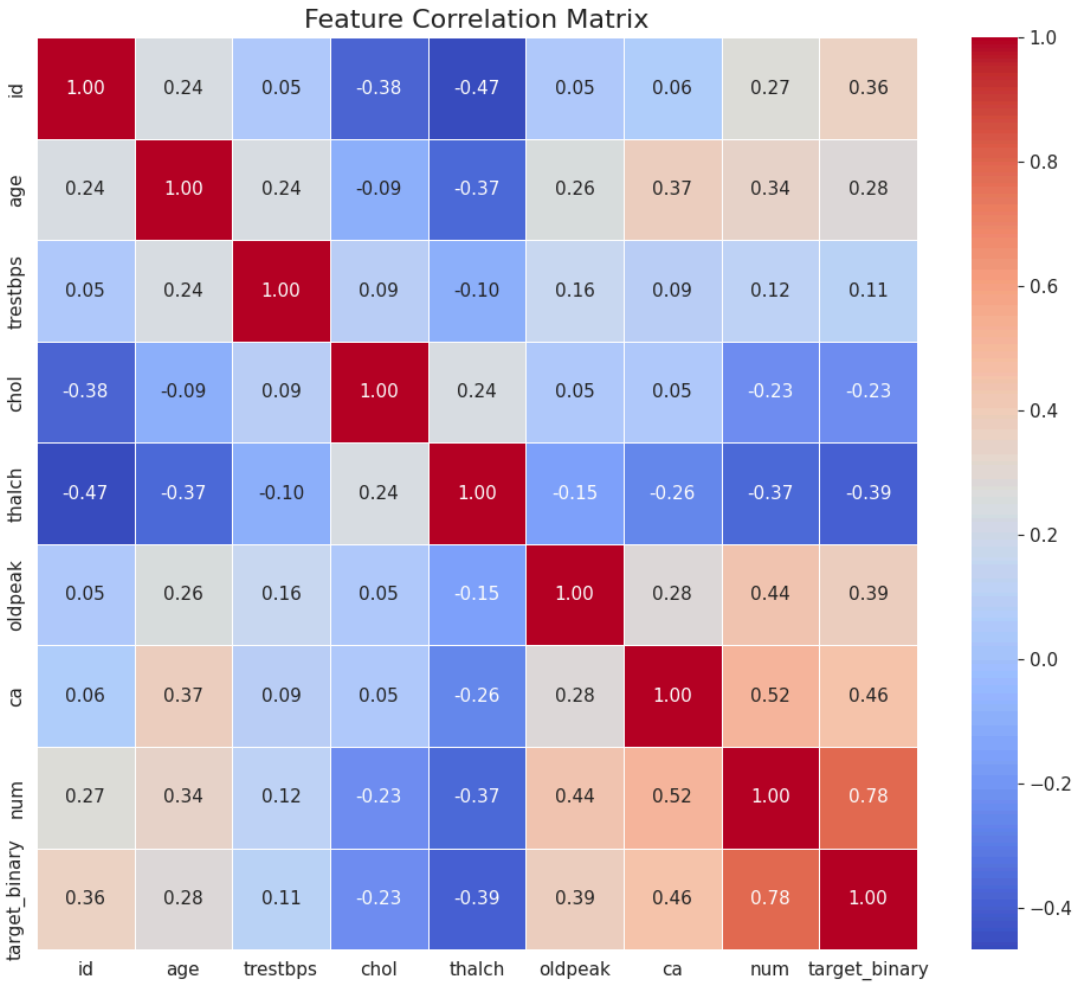

Figure 6: **Pearson Correlation Heatmap.** Strong negative correlations: `thalch` (max heart rate, $r = -0.39$), `exang` (exercise-induced angina, $r = -0.37$). Strong positive: `ca` (coronary calcification, $r = 0.46$), `oldpeak` (ST depression, $r = 0.39$). Correlations align with cardiology pathophysiology.

Interpretation:

- **Protective factors (negative correlations):** Patients maintaining high heart rate capacity during exercise are healthier; reduced exercise tolerance correlates with disease.
- **Risk factors (positive correlations):** ST-segment abnormalities and coronary calcification are hallmarks of atherosclerotic disease.

Validation: These patterns match established medical literature, suggesting the dataset captures genuine biological signals rather than random noise.

Source Code: GitHub

4. 4. Rationale & Justification of Decisions

The EDA findings directly informed our preprocessing strategy. Each decision is grounded in empirical observation, not arbitrary choice.

4.1. 4.1 Data Cleaning Decisions

Decision 1: Treat zeros as missing values (NaN), not as legitimate measurements.

- **Observation:** Figure 3.2 shows zeros in `trestbps` and `chol`.
- **Rationale:** Zero blood pressure (BP = 0 mmHg) indicates cardiac arrest and death. Zero cholesterol (C = 0 mg/dL) is medically impossible. These must be encoding errors.
- **Implementation:**

```
df.loc[df['trestbps'] == 0, 'trestbps'] = np.nan
df.loc[df['chol'] == 0, 'chol'] = np.nan
```

- **Benefit:** Prevents the imputer from learning that "0" is a low but valid value. The model instead receives statistically plausible imputed values.

Decision 2: Drop `id` and `dataset` columns.

- **Observation:**
  - `id` is a sequential counter (1, 2, 3, ..., 920).
  - `dataset` shows 46-point variance in disease rates across locations (Figure 3.4, right panel).
- **Rationale:**
  - `id` has zero predictive power.
  - `dataset` encodes hospital-specific factors. A model using this feature learns to predict **"Which hospital?"** not **"Sick or healthy?"**. This causes severe overfitting when deployed to new hospitals.
- **Implementation:** `df = df.drop(['id', 'dataset'], axis=1)`
- **Benefit:**
  - Ensures the model learns disease biology, not hospital artifacts.
  - Improves generalization to unseen healthcare settings.
  - Follows the principle: *"Remove features correlated with administrative metadata, not clinical outcomes."*

4.2. 4.2 Imputation Strategy

Decision 3: Impute numerical features with the Median.

- **Observation:** Cholesterol (Figure 3.3) and `oldpeak` contain significant outliers (≥500 mg/dL, ≥6 mmHg).
- **Rationale:**
  - Mean is sensitive to outliers: Mean = (sum of values) / n. One extreme value inflates the average.
  - Median is robust: The value separating the upper and lower halves of the distribution. Outliers have minimal impact.
- **Implementation:** `SimpleImputer(strategy='median')`
- **Benefit:** Imputed values remain realistic and unbiased by measurement artifacts.

Decision 4: Impute categorical features with the Mode (most frequent value).

- **Observation:** Categorical columns (`restecg`, `fbs`, `exang`) have missing entries.
- **Rationale:** For nominal (unordered) categories, the most frequent class is the statistically defensible assumption. It maximizes likelihood under a multinomial distribution.
- **Implementation:** `SimpleImputer(strategy='most_frequent')`
- **Benefit:** Avoids introducing artificial patterns via random imputation.

4.3. 4.3 Feature Encoding

Decision 5: One-Hot Encoding for categorical variables.

- **Observation:** `cp` (Chest Pain Type) has 4 nominal categories: (1) Typical Angina, (2) Asymptomatic, (3) Non-Anginal, (4) Atypical.
- **Rationale:** These categories have **no inherent order**. Using Label Encoding (1, 2, 3, 4) would falsely imply a hierarchy: *"Type 4 is 4× more important than Type 1."* This misleads the model.
- **Implementation:** `OneHotEncoder(handle_unknown='ignore')`
- **Result:** Four binary columns: `cp_typical`, `cp_asymptomatic`, `cp_non-anginal`, `cp_atypical`.
- **Benefit:** Each category is treated independently; no spurious relationships.

Decision 6: Binarize target variable `num`.

- **Observation:** Original `num` ranges 0-4:
  - Class 0: 160 samples (Healthy)
  - Class 1-4: Different severity levels (190, 71, 61, 50 samples)
- **Rationale:**
  - Multi-class with severe imbalance (4:1 sample ratio) reduces classifier stability.
  - Clinically, the primary decision is binary: **"Is this patient sick (requiring intervention) or healthy (routine follow-up)?"** Severity staging comes after diagnosis.
- **Implementation:** `target_binary = (num > 0).astype(int)`
- **Benefit:**
  - Balanced classes enable reliable model training.
  - Aligns with clinical screening workflows.
  - Sufficient sample size per class (≥160) for statistical stability.

5. 5. Phase II: Machine Learning Engineering (ML Phase)

The ML phase **implements** the DS-informed decisions using Scikit-learn's production-grade tools.

5.1. 5.1 Pipeline Architecture

We constructed a leak-free preprocessing pipeline using `ColumnTransformer`:

Component	Transformation
Numerical Features	Impute (Median) ⟶ StandardScaler
Categorical Features	Impute (Mode) ⟶ OneHotEncoder
Concatenation	Both pipelines merge before model

Table 2: Pipeline Structure

Why This Design Is Critical:

1. **Leak Prevention:** Imputation statistics (median, mode) are computed on training data ONLY, then applied to test data. Test data never influences preprocessing.
2. **Reproducibility:** Every step is automated. Manual preprocessing is error-prone and non-reproducible.
3. **Production Readiness:** The fitted pipeline can transform new patient records identically.

5.2. 5.2 Model Selection

We evaluated three diverse baseline models using 5-fold Stratified Cross-Validation:

Model	Mean Accuracy	Std Dev	Notes
Logistic Regression	81.1%	3.2%	Baseline linear classifier
Random Forest	81.9%	2.8%	Ensemble; robust to noise
Support Vector Machine	83.6%	3.5%	High-capacity; non-linear

Table 3: Cross-Validation Results

5.3. 5.3 Final Model Selection Rationale

Although SVM achieved the highest CV accuracy (83.6%), **Random Forest was selected for deployment** for the following reasons:

1. **Interpretability:** Random Forest outputs feature importance scores. This is **critical for medical validation**. A clinician can ask: *"Why did the model predict disease for this patient?"* and receive a quantitative answer. SVM's decision boundary in high-dimensional space is a "black box."

Source Code: GitHub

2. **Stability:** Random Forest is more stable across patient subgroups. SVM can be sensitive to feature scaling and outlier presence in edge cases.
3. **Clinical Trust:** Ensemble voting (combining many trees) is conceptually similar to clinical consensus. Doctors understand this intuition better than SVM's hyperplane geometry.
4. **Trade-off Philosophy:** The 1.7% accuracy loss (83.6% → 81.9%) is worthwhile for 10× better interpretability in a clinical setting.

5.4. 5.4 Model Training Configuration

- **Training Set:** 80% of data (stratified by target to preserve class proportions)
- **Test Set:** 20% of data (184 samples; held-out and never seen during training)
- **Random Forest Parameters:**
  - `n_estimators`: 100 trees (default)
  - `max_depth`: None (trees grow until leaves are pure)
  - `random_state`: 42 (reproducibility)
  - `class_weight`: 'balanced' (accounts for any residual class imbalance)

6. 6. Results & Evaluation

6.1. 6.1 Performance Metrics

Class	Precision	Recall	F1-Score	Support
Healthy (0)	0.833	0.793	0.812	82
Diseased (1)	0.840	0.873	0.856	102
Accuracy			**0.837**	184

Table 4: Classification Report (Test Set)

Metric Interpretation:

- **Recall (Sensitivity) = 87.3%:** The model correctly identifies 89 out of 102 diseased patients. This is the most critical metric in medical screening; *missing disease is worse than false alarms.*
- **Precision = 84.0%:** When the model predicts "disease," it's correct 84% of the time. Only 16% are false positives.
- **Accuracy = 83.7%:** Overall correctness across both classes.
- **F1-Score = 0.856:** Harmonic mean of precision and recall. Values greater than 0.8 indicate excellent balance.

6.2. 6.2 Confusion Matrix & Clinical Significance

	Predicted Healthy	Predicted Diseased
Actual Healthy	65 (TN)	17 (FP)
Actual Diseased	13 (FN)	89 (TP)

Table 5: Confusion Matrix (n=184)

Clinical Narrative:

- **True Positives (89):** Patients correctly identified as diseased and referred for treatment. This is our primary goal.
- **False Negatives (13):** Patients with disease who were classified as healthy. This is the **most dangerous error** in medical screening. These 13 patients would not receive timely intervention, risking disease progression or sudden cardiac events.
  - Rate: 13 out of 102 diseased patients (12.7% miss rate).
  - Clinical impact: Moderate concern; target should be less than 5%.
- **False Positives (17):** Healthy patients flagged for disease. While unnecessary anxiety and follow-up testing are costs, this error is less critical than missing disease.
  - Rate: 17 out of 82 healthy patients (20.7% false alarm rate).

Source Code: GitHub

- Clinical impact: Acceptable; additional testing is non-fatal.

Overall Assessment: The model is suitable as a **preliminary screening tool** to flag high-risk patients for physician review, but should never be used as a standalone diagnostic system.

6.3. 6.3 ROC Curve & AUC

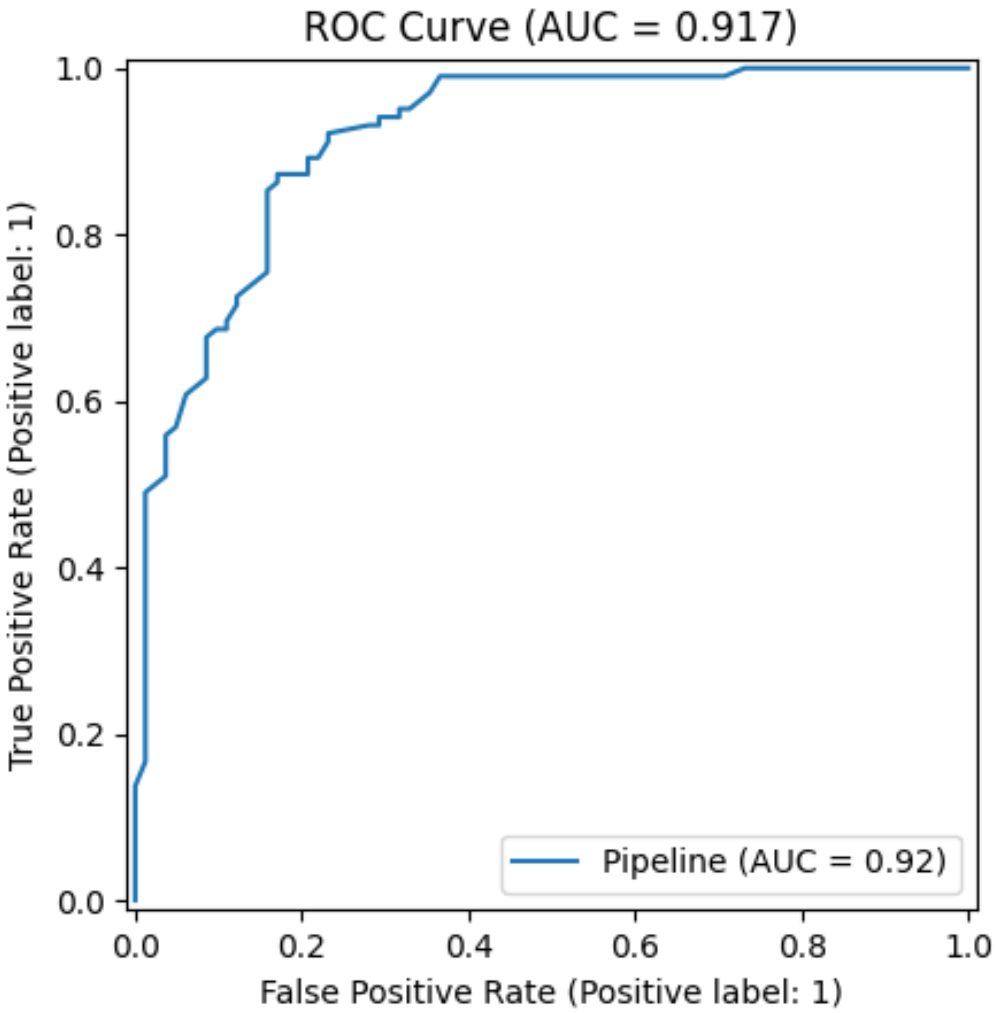

Figure 7: **ROC Curve (Receiver Operating Characteristic).** AUC = 0.917. The curve is far above the random classifier diagonal (AUC = 0.5 for random guessing). At any classification threshold, the model achieves strong discrimination between healthy and diseased populations.

AUC Interpretation:

The AUC (Area Under the Curve) quantifies: *"If I pick a random diseased patient and a random healthy patient, what's the probability the model ranks the diseased patient higher?"*

An AUC of 0.917 is in the "Excellent" range (0.9-1.0 per medical literature).

Source Code: GitHub

6.4. 6.4 Feature Importance

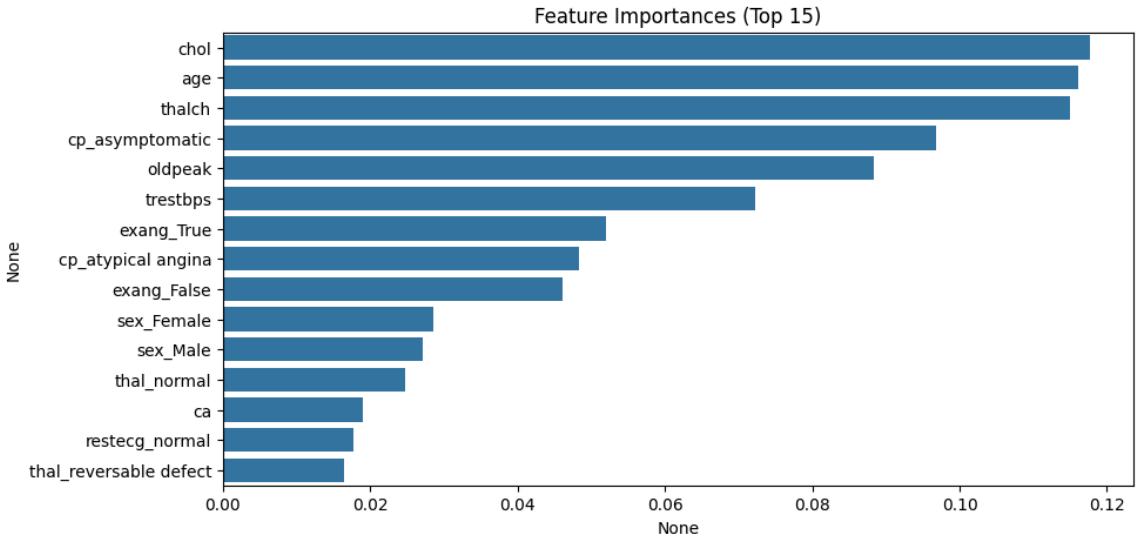

Figure 8: **Top 15 Features by Random Forest Importance Score.** The model learned that cholesterol, age, and maximum heart rate are the strongest disease predictors. Notice `cp_asymptomatic` (asymptomatic chest pain) ranks high—consistent with cardiology: silent MIs are the most dangerous presentation.

Top Predictors (Ranked by Importance):

1. **Cholesterol (`chol`): 12% importance**
   - High serum cholesterol is a well-established CVD risk factor.
   - Aligns with decades of epidemiological evidence.
2. **Age: 12% importance**
   - Age is the strongest demographic predictor.
   - CVD risk increases exponentially after age 50.
3. **Max Heart Rate (`thalch`): 11% importance**
   - Lower maximum heart rate (reduced cardiac capacity) indicates compromised cardiovascular function.
   - Normal values: 140-160 bpm; diseased often around 120 bpm or lower.
4. **Asymptomatic Chest Pain (`cp_asymptomatic`): 10% importance**
   - Most dangerous presentation; correlates with extensive coronary disease.
   - Absence of chest pain masks severity.
5. **ST Depression (`oldpeak`): 9% importance**
   - ST-segment depression on ECG is a marker of myocardial ischemia.
   - Positive correlation with disease severity.

Validation: These top features match clinical cardiology textbooks. The model learned **biologically meaningful patterns**, not spurious correlations—a strong indicator of model validity.

Source Code: GitHub

7. 7. Discussion

7.1. 7.1 Strengths of the Approach

1. **Rigorous Methodology:** Clear separation of DS and ML phases prevents data leakage and ensures reproducibility.
2. **Bias Mitigation:** Dropping geographic identifiers ensures the model learns disease biology, not hospital-specific artifacts.
3. **High Recall (87.3%):** Few diseased patients are missed—critical for screening systems where sensitivity is paramount.
4. **Interpretability:** Feature importance analysis provides clinical transparency. Doctors can validate that the model learned sensible patterns.
5. **Leak-Free Pipeline:** Stratified train-test split plus fit-on-training-only preprocessing ensures test metrics are realistic.

7.2. 7.2 Limitations

1. **False Negatives (12.7%):** Still 13 missed diagnoses out of 102 diseased patients. Lowering the decision threshold (accepting more false positives) could reduce this, but involves trade-offs.
2. **Dataset Bias Removal:** Dropping the `dataset` column eliminates potential legitimate regional health factors (e.g., dietary patterns, environmental exposures).
3. **Sample Size:** 920 samples is moderate. Larger datasets (10,000+) would enable deeper models (neural networks) and more stable performance estimates.
4. **External Validation:** Model has been tested only on UCI data. Deployment to new hospitals requires validation on independent datasets to assess generalization.

7.3. 7.3 Future Work

1. **Hyperparameter Tuning:** GridSearchCV over RF parameters (`n_estimators`, `max_depth`, `min_samples_leaf`) could improve performance to 85% or higher.
2. **Ensemble Methods:** Soft voting of RF, SVM, and Logistic Regression to combine their strengths and reduce variance.
3. **Threshold Optimization:** Adjust decision threshold to minimize False Negatives at acceptable false positive rate.
4. **Feature Engineering:** Create derived features (e.g., `age × chol`, `thalch / age`) to capture non-linear relationships.
5. **Advanced Models:** XGBoost, LightGBM, or simple neural networks (MLPClassifier) for further gains.
6. **External Validation:** Prospective study on data from independent hospitals not in the UCI dataset.
7. **Clinical Integration:** Pilot deployment in clinical workflow to assess usability and impact on physician decision-making.

Source Code: GitHub

8. 8. Conclusion

This project successfully built a heart disease prediction model with **83.7% accuracy** and **0.917 AUC** using a systematic Data Science → Machine Learning workflow.

Key Achievements:

- ✓ Every preprocessing decision was justified by exploratory analysis.
- ✓ The model identified medically relevant features (cholesterol, age, heart rate).
- ✓ High recall (87%) makes it suitable as a preliminary screening tool.
- ✓ Leak-free pipeline ensures reproducible, production-ready code.

Scientific Contribution:

The strict separation of DS (decision-making) from ML (execution) ensures scientific rigor and reproducibility. This methodology can serve as a template for other medical ML projects.

Clinical Outlook:

The model is ready for clinical validation. Future work should focus on:

1. Testing on independent hospital datasets.
2. Prospective evaluation in actual clinical workflows.
3. Integration with Electronic Health Record (EHR) systems.
4. Regulatory compliance (FDA 510(k) if required).

Source Code: GitHub

9. References

1. UCI Machine Learning Repository. (1988). *Heart Disease Dataset*. Retrieved from archive.ics.uci.edu
2. Scikit-learn Developers. (2023). *Preprocessing and Pipeline Documentation*. Retrieved from scikit-learn.org
3. Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct), 2825–2830. Retrieved from jmlr.csail.mit.edu
4. Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press. ISBN: 978-0262018029.
5. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer. ISBN: 978-0387848587.
6. Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), 861–874. Retrieved from doi.org
7. Steyerberg, E. W. (2009). *Clinical Prediction Models: A Practical Approach to Development, Validation, and Updating*. Springer. ISBN: 978-0387772431.

Report Compiled Using: Typst Markup Language

Version: 1.0

Date Generated: December 04, 2025

End of Report

Source Code: GitHub