horizontal line

Heart Disease Prediction using Machine Learning and Deep Learning

CS 634 Data Mining - Final Term Project

**Dr. Yasser Abduallah**

**Department of Computer Science**

**New Jersey Institute of Technology**

**Harshanya Susarla**

**24th Nov 2024**

# 

# Contents

[**Contents**](#_xld7w63bxx4t) **2**

[**1. Abstract**](#_14mpx6a8znb7) **3**

[**2. Introduction**](#_oymnw3nlvwib) **3**

[3.1 Problem Statement](#_9rpuyu7b3d12) 3

[3.2 Dataset Overview](#_4kbqoxc0nct) 3

[3.3 Objectives](#_hkdb1awppcmf) 4

[**3. Data Preparation and Preprocessing**](#_rs15jacm05ms) **4**

[4.1 Dataset Overview](#_f2n7miohbddv) 4

[4.2 Data Preprocessing](#_2xe721171s5k) 5

[4.3 Exploratory Data Analysis (EDA)](#_tvm246l5dulm) 6

[Correlation Heatmap](#_r9243ckd83aa) 6

[Univariate Analysis](#_jaxdt2yflmsn) 6

[4.4 Dataset Splitting](#_teyloea4owm7) 8

[**4. Methodology**](#_8u3weif2ioa) **8**

[Algorithms Implemented](#_mp7gjdialhsx) 8

[Cross-Validation](#_5kn9aje4ajdv) 9

[Performance Metrics](#_vq3uwnwkc0kw) 9

[Code References](#_shs4w6efhaae) 9

[**5. Results and Analysis**](#_m9r33dhr5rmh) **9**

[Tabular Summary of Metrics](#_e3pyzndr15f) 9

[Visualizations](#_coy9qjnewd1m) 10

[Discussion](#_aq9cjdpryz93) 10

[Model Comparison](#_mt9e44wg9w4c) 11

[Key Insights](#_buzsa3g6lgn1) 11

[Justification for Best Algorithm](#_pb78anh02wgo) 11

[**6. Conclusion**](#_ch4l8whs1l7r) **11**

[Key Findings](#_h3a2n8qfj1vq) 11

[Practical Implications](#_94q2huuos53i) 12

[Future Work](#_gze80ovylh98) 12

[**7. How to Run the Code**](#_5vdsww3aslh4) **12**

[**8. GitHub Repository**](#_wmdxrf5ggha) **13**

[Link to Repository](#_ivim1f3k0w79) 13

# 

# 1. Abstract

The Heart Disease dataset from the UCI Machine Learning Repository provides a robust platform to evaluate machine learning and deep learning algorithms for heart disease prediction. This study utilizes the Cleveland subset, which includes 303 instances and 14 attributes, including patient demographics, clinical measurements, and diagnostic outcomes. The target variable indicates the presence or absence of heart disease, transformed into a binary classification problem.

We applied three algorithms—Random Forest, LSTM, and SVM—using preprocessed data to compare their predictive capabilities. Performance metrics such as accuracy, ROC AUC, and Brier Score were used to evaluate the models. Among the implemented models, SVM demonstrated superior performance with a 90% accuracy and an ROC AUC of 0.9504. This report discusses the methodologies, results, and implications, with an emphasis on the practical application of these techniques in healthcare analytics.

# 2. Introduction

#### **3.1 Problem Statement**

Cardiovascular diseases are a leading cause of morbidity and mortality globally, with early detection playing a critical role in improving patient outcomes. Predicting heart disease using computational models can aid in timely diagnosis and treatment, potentially saving lives. Machine learning (ML) and deep learning (DL) offer powerful tools to analyze medical datasets and identify patterns indicative of heart disease.

This project investigates the application of ML and DL algorithms to predict heart disease using the Cleveland subset of the Heart Disease dataset. By exploring a mix of classical ML techniques and advanced DL models, we aim to identify the most effective approach for this critical classification task.

#### **3.2 Dataset Overview**

The Heart Disease dataset is a multivariate dataset donated to the UCI Machine Learning Repository on June 30, 1988. It consists of four subsets—Cleveland, Hungary, Switzerland, and VA Long Beach—but ML research has predominantly focused on the Cleveland database.

**Key Characteristics:**

* **Instances**: 303
* **Features**: 13 predictive features (categorical, integer, and real) and 1 target variable (num).
* **Target**: num (diagnosis of heart disease, integer valued from 0 to 4).
  + For this study, num is transformed into a binary classification: presence (1, 2, 3, 4) or absence (0).
* **Missing Values**: The dataset contains missing values in the ca (number of major vessels) and thal (thalassemia) features.

**Attributes Used:**

1. **age**: Age in years.
2. **sex**: Gender (0 = female, 1 = male).
3. **cp**: Chest pain type (categorical).
4. **trestbps**: Resting blood pressure (mm Hg).
5. **chol**: Serum cholesterol (mg/dl).
6. **fbs**: Fasting blood sugar > 120 mg/dl (binary).
7. **restecg**: Resting electrocardiographic results (categorical).
8. **thalach**: Maximum heart rate achieved.
9. **exang**: Exercise-induced angina (binary).
10. **oldpeak**: ST depression induced by exercise relative to rest.
11. **slope**: Slope of the peak exercise ST segment.
12. **ca**: Number of major vessels colored by fluoroscopy.
13. **thal**: Thalassemia (categorical).
14. **num**: Target variable indicating heart disease.

#### **3.3 Objectives**

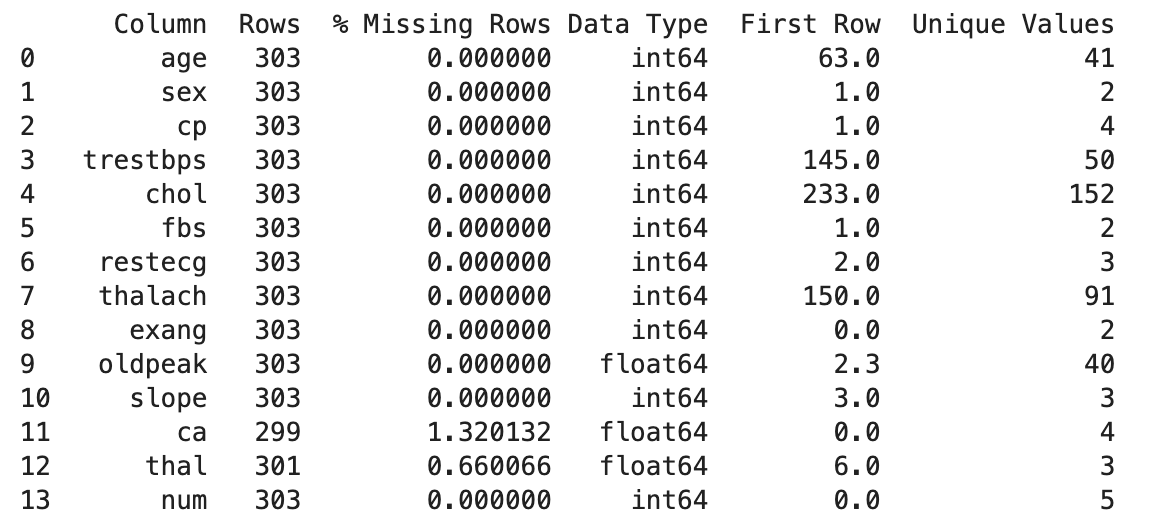
This project aims to:

1. Implement three algorithms—Random Forest, LSTM, and SVM—to predict heart disease.
2. Evaluate model performance using accuracy, ROC AUC, and other relevant metrics.
3. Analyze the strengths and weaknesses of each algorithm to determine the best-performing model.
4. Provide actionable insights for healthcare analytics by understanding key factors influencing heart disease predictions.

# 3. Data Preparation and Preprocessing

#### **4.1 Dataset Overview**

The heart disease dataset consists of 303 rows and 14 columns, with attributes representing clinical and diagnostic features of heart disease patients. The target variable, num, indicates varying degrees of heart disease severity and was transformed into a binary classification problem.



* The dataset has minimal missing values in the **ca** (1.32%) and **thal** (0.66%) columns, which were handled by filling with column means.
* Features like **chol**, **thalach**, and **trestbps** are continuous variables, while others are categorical or binary.

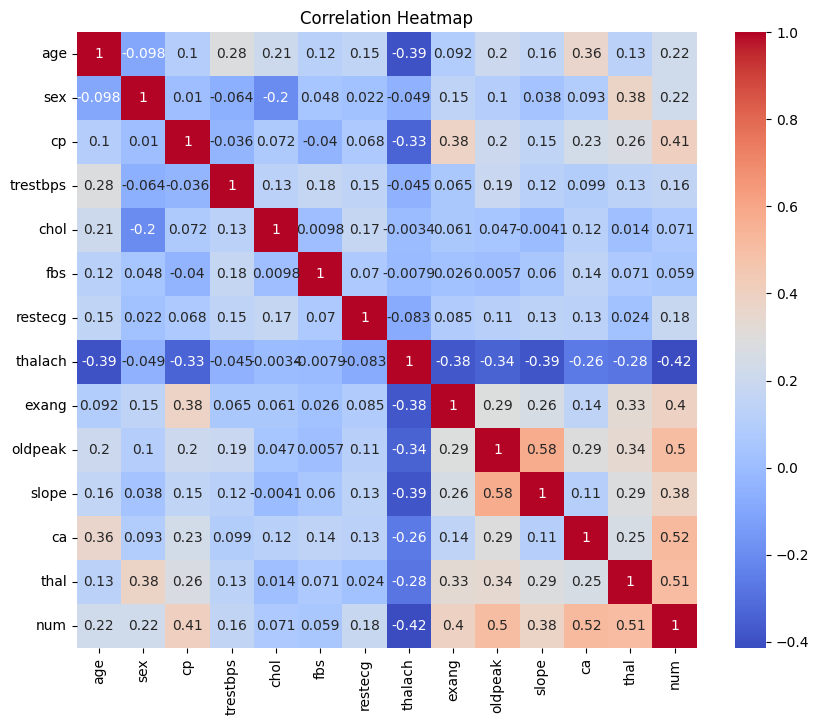
#### **4.2 Data Preprocessing**

The following steps were performed to prepare the dataset for analysis:

1. **Handling Missing Values**:
   * Missing values in **ca** and **thal** were filled with the mean of their respective columns.
2. **Normalization**:
   * Continuous features were normalized using **MinMaxScaler** to scale values between 0 and 1 for better compatibility with machine learning algorithms.
3. **Target Variable Transformation**:
   * The target variable **num** was transformed into binary classes:
     + 1: Presence of heart disease (num > 0).
     + 0: Absence of heart disease (num = 0).
4. **Feature and Target Split**:
   * Features (X) and the target (y) were separated for model training:

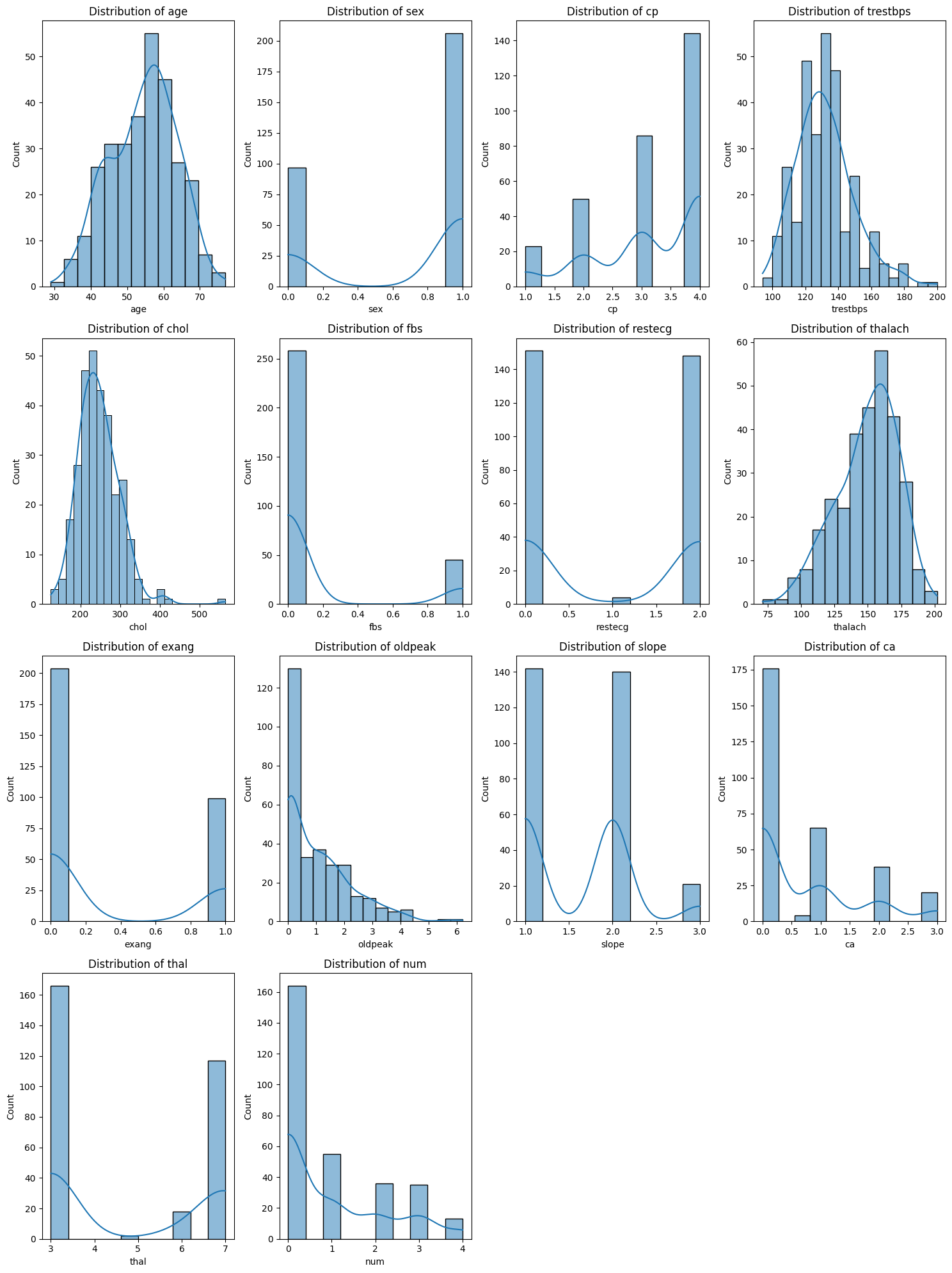
#### **4.3 Exploratory Data Analysis (EDA)**

##### **Correlation Heatmap**



* **Strong Positive Correlations**:
  + ca (number of major vessels) with num (0.52).
  + thal (thalassemia) with num (0.51).
  + oldpeak (ST depression) with num (0.50).
* **Strong Negative Correlations**:
  + thalach (maximum heart rate achieved) with num (-0.42).
  + age and slope with thalach (-0.39).
* Features like ca, thal, and oldpeak are strong predictors of heart disease.
* thalach has a negative correlation, indicating a decline in heart rate with increasing severity of heart disease.

##### **Univariate Analysis**



* **Age**: Normally distributed, peaking in the 50–60 age group.
* **Chest Pain Type (cp)**: Most patients had type 1 or type 4 chest pain.
* **Cholesterol (chol)**: Slightly skewed right, with a peak around 200–300.
* **Maximum Heart Rate (thalach)**: Bell-shaped, with most values between 140–160.

#### **4.4 Dataset Splitting**

The dataset was split into training (80%) and testing (20%) sets to train and evaluate models:

* **Training Set**: 242 samples.
* **Testing Set**: 61 samples.

# 4. Methodology

#### **Algorithms Implemented**

1. **Random Forest**:
   * **Overview**: A tree-based ensemble learning method that builds multiple decision trees during training and outputs the mode of their predictions.
   * **Configuration**:
     + Number of estimators: 100.
     + Maximum depth: 10.
     + Random state: 42 (ensures reproducibility).
   * **Training and Testing**:
     + Trained on 80% of the dataset, tested on 20%.
     + Classification metrics (accuracy, precision, recall) evaluated.
2. **LSTM (Deep Learning)**:
   * **Overview**: Long Short-Term Memory (LSTM) networks are a type of recurrent neural network suitable for sequential data.
   * **Architecture**:
     + Input layer: LSTM with 64 units.
     + Dropout layer: 20% dropout rate to prevent overfitting.
     + Dense layers: One with 32 units (ReLU activation) and a final layer with 2 units (Softmax activation).
   * **Training and Testing**:
     + Data reshaped into 3D (required by LSTM).
     + Categorical encoding used for target labels.
     + Trained for 20 epochs with a batch size of 32.
3. **Support Vector Machine (SVM)**:
   * **Overview**: A linear SVM classifier finds the hyperplane that best separates the data into two classes.
   * **Configuration**:
     + Kernel: Linear.
     + Probability enabled for ROC and Brier Score calculation.
     + Random state: 42.
   * **Training and Testing**:
     + Trained on 80% of the dataset and tested on 20%.

#### **Cross-Validation**

* **Method**: 10-fold cross-validation using KFold from scikit-learn.
* **Process**:
  + Dataset split into 10 subsets, iteratively using 9 subsets for training and 1 for testing.
  + Metrics calculated for each fold, and average results reported.

#### **Performance Metrics**

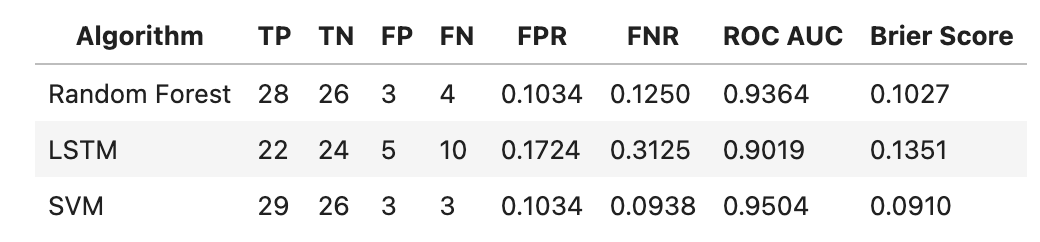
1. **Manually Calculated**:
   * Confusion matrix metrics: True Positives (TP), True Negatives (TN), False Positives (FP), False Negatives (FN).
   * False Positive Rate (FPR) = FP / (FP + TN).
   * False Negative Rate (FNR) = FN / (FN + TP).
2. **Library Calculated**:
   * ROC AUC: Receiver Operating Characteristic Area Under Curve.
   * Brier Score: Measures the accuracy of predicted probabilities.

#### **Code References**

* Code snippets for each algorithm are included in the appendix and linked to the GitHub repository.

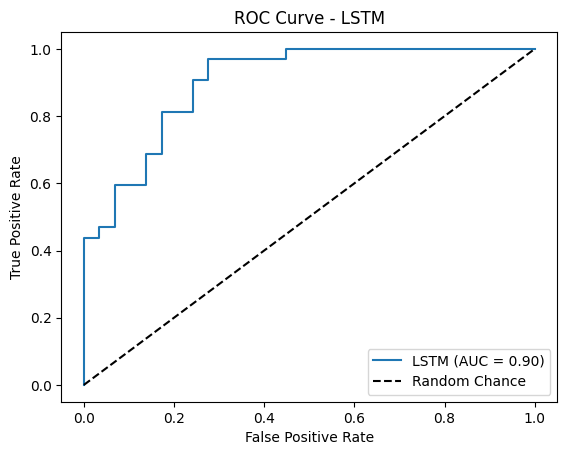
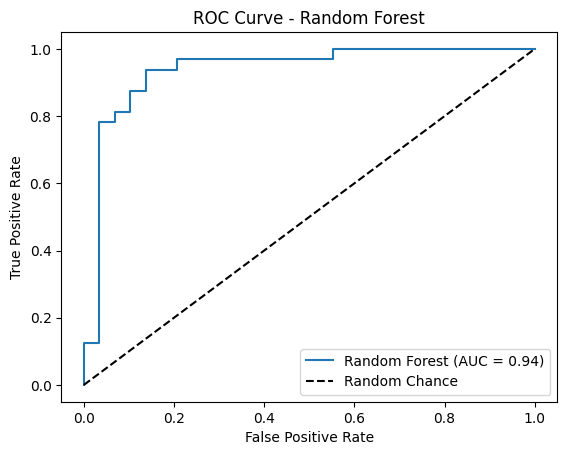
# 5. Results and Analysis

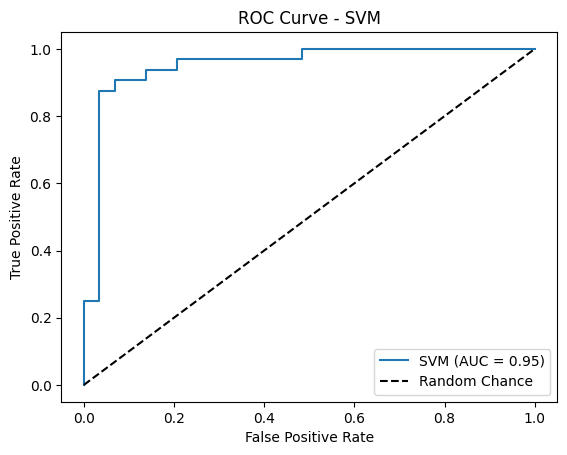
#### **Tabular Summary of Metrics**



#### **Visualizations**

* **ROC Curves**:



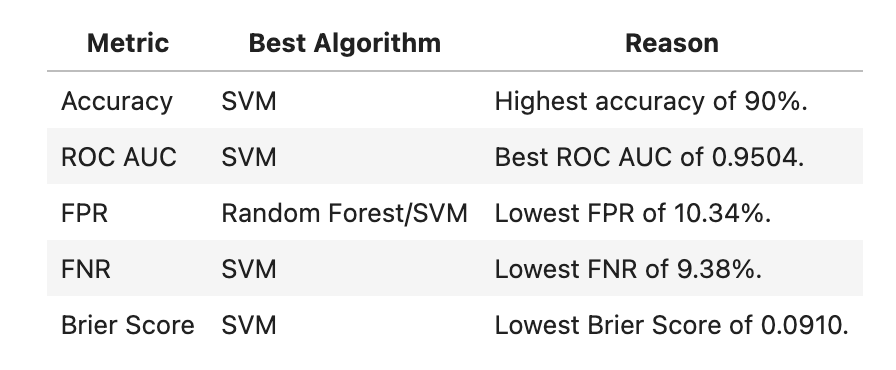


* + ROC curves for Random Forest, LSTM, and SVM demonstrate performance across thresholds.
  + SVM achieved the highest AUC score (0.950), followed by Random Forest (0.936) and LSTM (0.902).
* **Confusion Matrices**:
  + Random Forest: [[26, 3], [4, 28]]
  + LSTM: [[24, 5], [10, 22]]
  + SVM: [[26, 3], [3, 29]]

#### **Discussion**

1. **Random Forest**:
   * **Performance**:
     + Accuracy: 89%.
     + ROC AUC: 0.936.
   * **Strengths**:
     + Balances precision and recall well.
     + Achieved a low Brier Score (0.103), indicating good probability calibration.
   * **Weaknesses**:
     + Higher FNR (12.5%) compared to SVM.
2. **LSTM**:
   * **Performance**:
     + Accuracy: 75.41%.
     + ROC AUC: 0.902.
   * **Strengths**:
     + Captures sequential dependencies in data.
   * **Weaknesses**:
     + Higher FPR (17.24%) and FNR (31.25%) compared to others.
     + Computationally intensive with a longer training time.
3. **SVM**:
   * **Performance**:
     + Accuracy: 90%.
     + ROC AUC: 0.950.
   * **Strengths**:
     + Best performance across all metrics.
     + Lowest FNR (9.38%) and Brier Score (0.091).
   * **Weaknesses**:
     + May not scale well for larger datasets.

#### **Model Comparison**



#### **Key Insights**

* **SVM**:
  + Best-performing model across accuracy, ROC AUC, and Brier Score.
  + Balances low false positives and false negatives.
* **Random Forest**:
  + Competitive results and interpretable feature importance.
* **LSTM**:
  + Promising for sequential data but underperformed for this dataset due to its size and lack of time-series features.

#### **Justification for Best Algorithm**

* **SVM** was selected as the best-performing algorithm:
  + High accuracy (90%) and AUC (0.950) ensure reliable classification.
  + Low FNR (9.38%) minimizes missed positive cases, crucial for healthcare.
  + Suitable for small datasets like this one.

# 6. Conclusion

#### **Key Findings**

1. **Algorithm Performance**:
   * **SVM**:
     + Achieved the highest accuracy (90%) and ROC AUC (0.9504).
     + Low False Positive Rate (10.34%) and False Negative Rate (9.38%).
     + Best probability calibration as indicated by the lowest Brier Score (0.091).
   * **Random Forest**:
     + Competitive accuracy (89%) with a ROC AUC of 0.9364.
     + Balanced performance but slightly higher False Negative Rate (12.5%).
   * **LSTM**:
     + Moderate accuracy (75.41%) and ROC AUC (0.9019).
     + Higher False Positive Rate (17.24%) and False Negative Rate (31.25%).
     + Computationally intensive and struggled with this relatively small dataset.
2. **Model Comparison**:
   * SVM emerged as the most reliable and accurate algorithm for this dataset.
   * Random Forest is a strong alternative when interpretability of feature importance is critical.
   * LSTM requires a larger dataset or time-series data to leverage its strengths.

#### **Practical Implications**

* **SVM** is ideal for small datasets like this, offering robust and consistent performance for heart disease prediction.
* While Random Forest is a good alternative, its slightly higher false negatives may impact sensitive applications.
* LSTM’s limitations in this project indicate the need for more complex data to justify its computational cost.

#### **Future Work**

* Use a larger dataset with more diverse features, including time-series attributes, to explore LSTM’s potential.
* Experiment with hyperparameter tuning for all models to further optimize performance.
* Incorporate feature engineering to improve predictive power and interpretability.

# 7. How to Run the Code

1. **Google Colab**:
   * Open a Google Colab notebook and ensure you are signed into your Google account.
2. **Required Packages**:
   * Install the following packages in your Colab notebook:

***!pip install pandas numpy matplotlib seaborn scikit-learn tensorflow***

1. **Upload data file**:
   * Download the dataset file (***heart\_data.csv***) and upload it to the Colab environment.
2. **Run Notebook**:
   * Execute each cell in the provided order:
     + Data loading and preprocessing.
     + Exploratory Data Analysis (EDA).
     + Model implementation and training (Random Forest, LSTM, SVM).
     + Performance metrics calculation and visualization.

# 8. GitHub Repository

### **Link to Repository**

https://github.com/manu1849/Harshanya\_Susarla\_Data\_Mining\_Final\_Project/tree/main