Documentation for ***Cardiovascular Disease*** Prediction Model

**Table of Contents**

[Introduction 2](#_Toc189517048)

[Dataset 2](#_Toc189517049)

[Data Preprocessing 2](#_Toc189517050)

[Categorical Encoding: 2](#_Toc189517051)

[Feature Scaling: 2](#_Toc189517052)

[Model Comparison 2](#_Toc189517053)

[Individual Model Performance 2](#_Toc189517054)

[Stacking Classifier Performance 3](#_Toc189517055)

[Visualizations 3](#_Toc189517056)

[ROC Curve Comparison 3](#_Toc189517057)

[Confusion Matrix 4](#_Toc189517058)

[Learning Curves 4](#_Toc189517059)

[Key Insights 5](#_Toc189517060)

[Limitations 5](#_Toc189517061)

[Future Improvements 6](#_Toc189517062)

[Conclusion 6](#_Toc189517063)

# Introduction

This document provides an overview of the machine learning model developed(in-progress till now) to predict cardiovascular diseases using the Cleveland Heart Disease dataset. The model employs various classification algorithms and evaluates their performance. The project is currently a work in progress, with plans for further improvements.

# Dataset

* **Source**: Cleveland Heart Disease dataset from Kaggle(<https://www.kaggle.com/datasets/ritwikb3/heart-disease-cleveland/data>). The dataset is the Cleveland Heart Disease dataset taken from the UCI repository.
* Original data: <https://archive.ics.uci.edu/ml/datasets/Heart+Disease>
* **Features**:
  + Demographic data (age, sex)
  + Medical measurements (blood pressure, cholesterol)
  + Clinical indicators (chest pain type, exercise-induced angina)
* **Target Variable**: Heart disease presence (binary classification)

# Data Preprocessing

## Categorical Encoding:

* + Mapped categorical variables to meaningful labels
  + Applied one-hot encoding

## Feature Scaling:

* + Standardized numerical features using StandardScaler
  + Scaled columns: age, blood pressure, cholesterol, max heart rate, ST depression

# Model Comparison

## Individual Model Performance

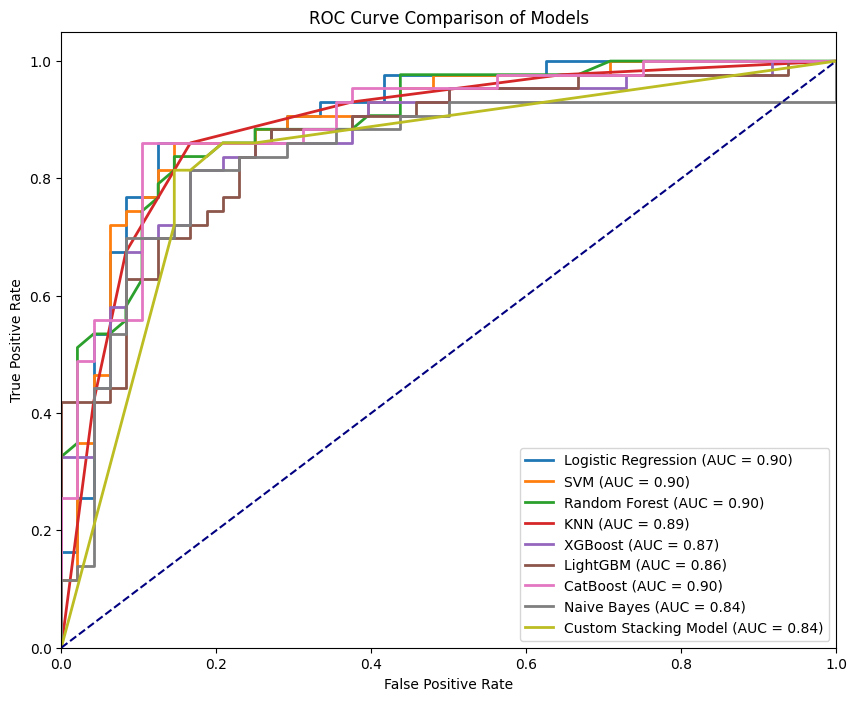
| **Model** | **Accuracy** | **ROC AUC** | **Key Characteristics** |
| --- | --- | --- | --- |
| Logistic Regression | 80.22% | 0.904 | Linear decision boundary |
| Support Vector Machine | 84.62% | 0.898 | Non-linear classification |
| Random Forest | 83.52% | 0.898 | Handles complex interactions |
| K-Nearest Neighbors | 84.62% | 0.890 | Proximity-based classification |
| XGBoost | 81.32% | 0.871 | Gradient boosting |
| LightGBM | 76.92% | 0.865 | Tree-based ensemble |
| CatBoost | 83.52% | 0.898 | Categorical feature handling |
| Naive Bayes | 64.84% | 0.837 | Probabilistic approach |

## Stacking Classifier Performance

* **Accuracy**: 83.52%
* **ROC AUC**: 0.838
* **Approach**: Combines predictions from Random Forest, XGBoost, LightGBM, and CatBoost
* **Final Estimator**: Logistic Regression

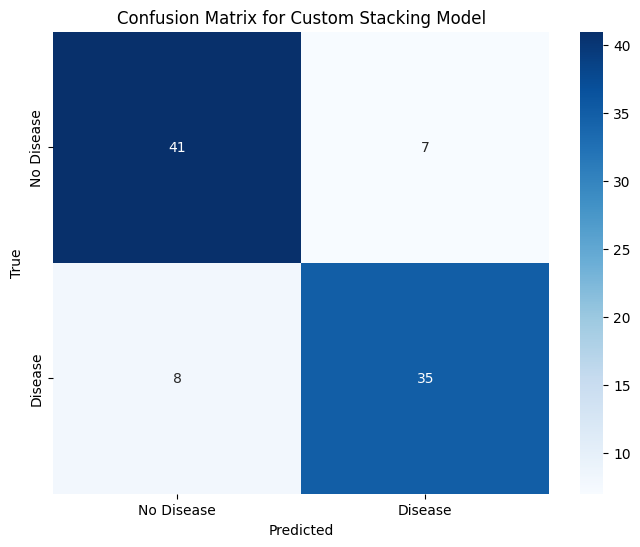
# Visualizations

## ROC Curve Comparison

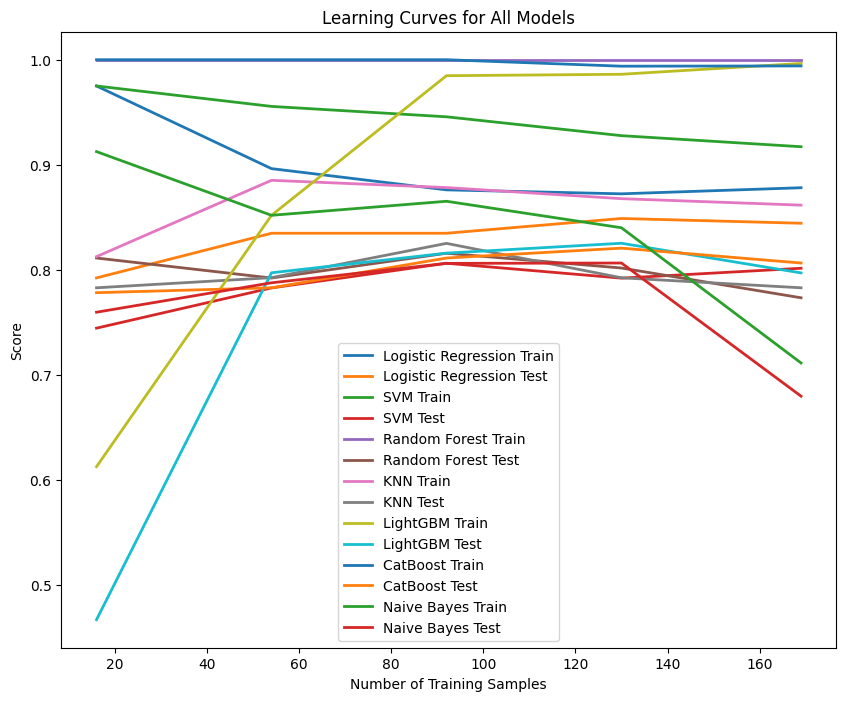
* + Compares model performance using Receiver Operating Characteristic curves
  + Shows trade-offs between true positive and false positive rates

## Confusion Matrix

* + Visual representation of model predictions
  + Highlights correct and incorrect classifications

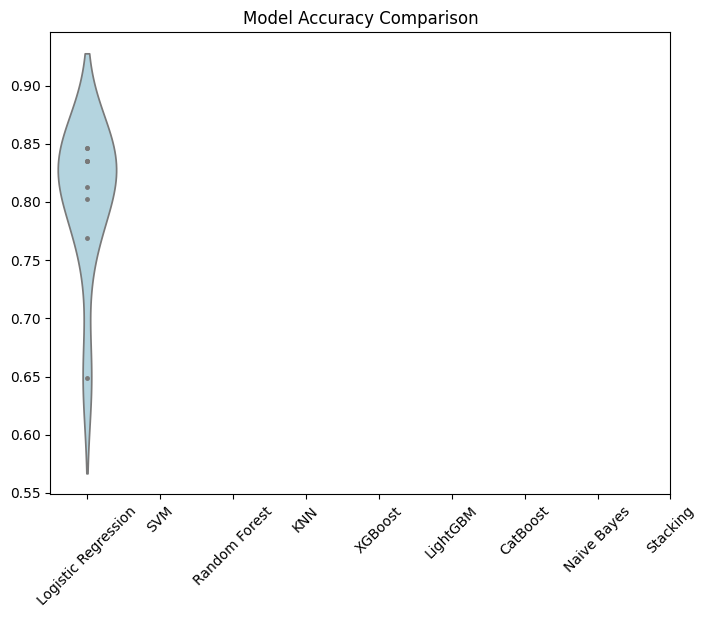


## Learning Curves

* + Demonstrates model performance with increasing training data
  + Helps understand model complexity and potential overfitting

# Key Insights

* SVM and KNN showed the highest accuracy (84.62%)
* Ensemble methods (Random Forest, CatBoost) performed consistently well
* Stacking classifier effectively combined multiple model strengths
* Naive Bayes showed the lowest performance, indicating non-probabilistic relationships in the data



# Limitations

* Relatively small dataset
* Potential class imbalance
* Limited to Cleveland dataset characteristics

# Future Improvements

Once the best model performance is achieved, I plan to develop a graphical user interface (GUI) that will allow users to input various information similar to that in the dataset. The GUI will utilize the trained model to provide predictions based on user input. Additionally, I aim to integrate conversational AI capabilities, potentially using OpenAI, Gemini or Claude to create a more interactive experience. This will enable the GUI to respond to user inquiries based on the model's predictions.

# Conclusion

This notebook demonstrates the process of building a cardiovascular disease prediction model using various machine learning algorithms. The stacking classifier achieved the best performance, showcasing the effectiveness of combining multiple models. The project is ongoing, with plans to develop a GUI for user interaction and predictions, along with conversational AI integration for enhanced user experience.