Capstone Project2

HEART DISEASE DETECTION USING CLASSIFICATION ALGORITHMS

Team Members

T120100312: Sharon A Dobbin
T120100316: Sherin Nayana B
T120100317: Prithvi Prabhu Pani V
T120100318: P Shreya
T120100324: Tejushree R
T120100330: Varshitha Sunnampalli

Table of Contents

Problem Statement

Goal

To build a **machine learning model** that can **predict the likelihood of heart disease** based on patient health indicators such as age, cholesterol, blood pressure, and chest pain type.

Problem Definition

Heart disease remains one of the leading causes of death worldwide. Early prediction can save lives, but clinical diagnosis is often time-consuming and resource-intensive.

The goal is to create an **automated, data-driven diagnostic tool** that can:

- Predict heart disease from patient data.
- Assist doctors in risk assessment.
- Improve accuracy and speed of detection.

Overview of the Project

This capstone project challenges students to develop and optimize machine learning models for predicting heart disease based on diagnostic test results.

Students will work with clinical diagnostic data to build classification models that can assist healthcare professionals in e arly detection and risk assessment

- **Data Understanding & Exploration**
  Load dataset and analyze structure, null values, and data types.

- **Data Preprocessing**
  Handle missing data, encode categorical features, scale numeric data.

- **Model Building**
  Train multiple ML models (Logistic Regression, Random Forest, SVM).

- **Model Evaluation**
  Use metrics: accuracy, precision, recall, F1-score, ROC-AUC.

- **Hyperparameter Tuning and Model Selection**
  Select optimal model.

Data generation & Loading

Heart Disease dataset contains **patient demographic, clinical, and diagnostic** information.

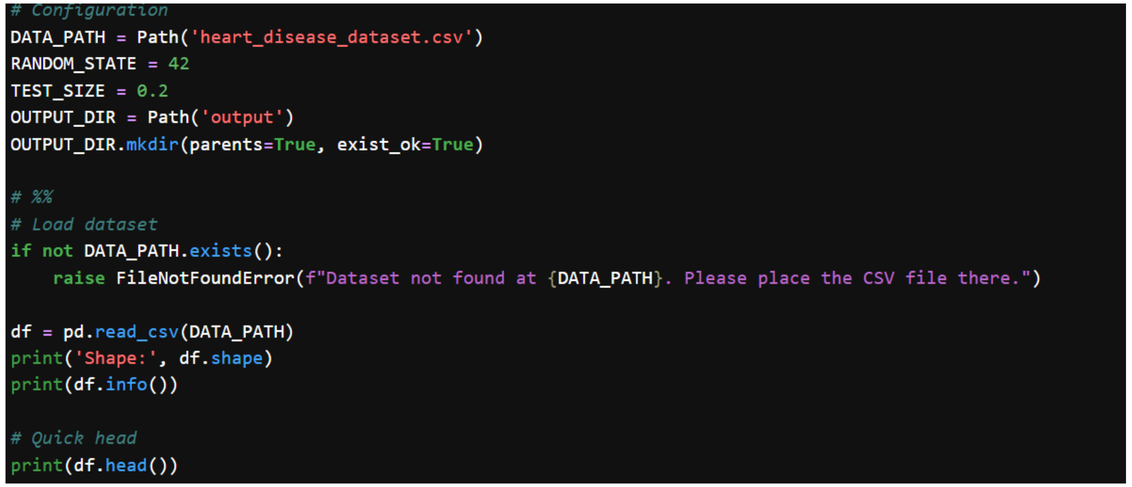
Target Variable: heart_disease → 0 = No disease, 1 = Disease present.

Features include:

- *Demographics:* age, sex
- *Clinical:* chest_pain_type, resting_blood_pressure, cholesterol, fasting_blood_sugar
- *Diagnostic Tests:* resting_ecg, max_heart_rate, exercise_induced_angina, st_depression, st_slope, num_major_vessels, thalassemia

- Dataset loaded using **Pandas** for analysis and exploration.

- Initial inspection includes checking **data types, missing values, and feature distributions**.

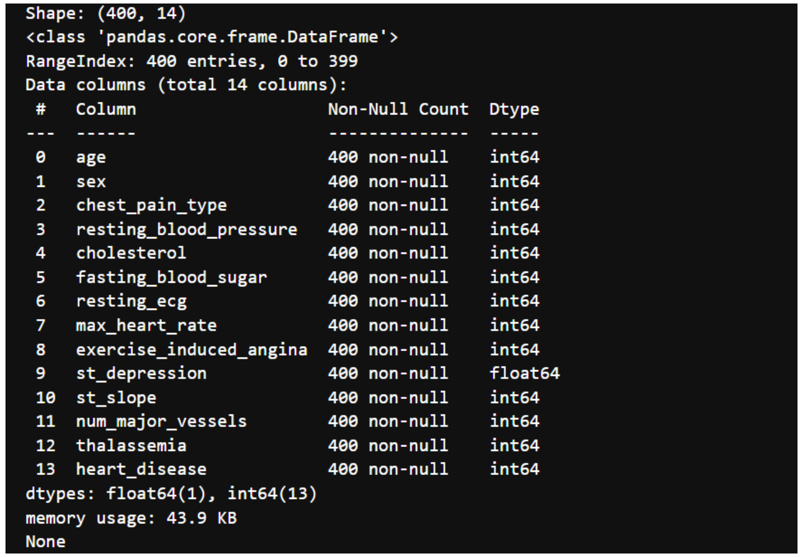


```
# Configuration
DATA_PATH = Path('heart_disease_dataset.csv')
RANDOM_STATE = 42
TEST_SIZE = 0.2
OUTPUT_DIR = Path('output')
OUTPUT_DIR.mkdir(parents=True, exist_ok=True)

# %%
# Load dataset
if not DATA_PATH.exists():
    raise FileNotFoundError(f"Dataset not found at {DATA_PATH}. Please place the CSV file there.")

df = pd.read_csv(DATA_PATH)
print('Shape:', df.shape)
print(df.info())

# Quick head
print(df.head())
```

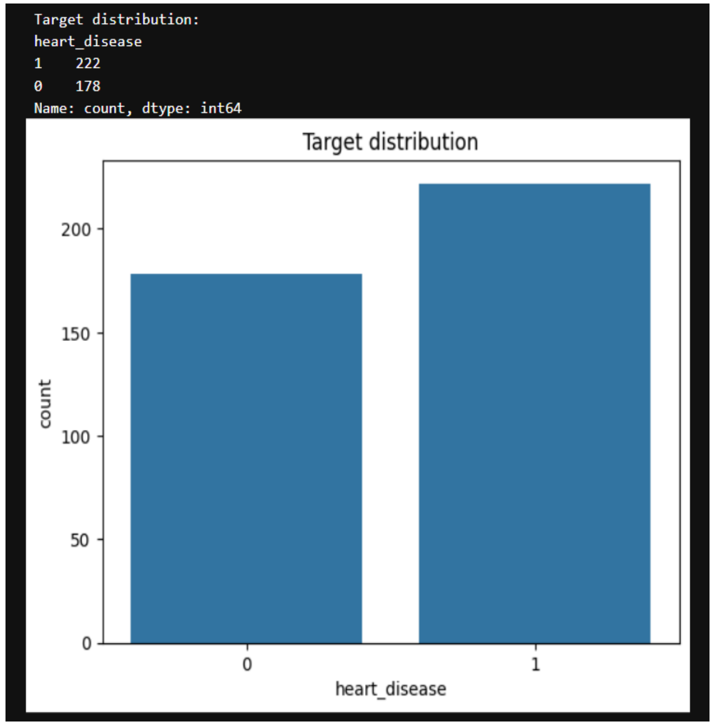


```
Shape: (400, 14)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 14 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   age                      400 non-null    int64
 1   sex                      400 non-null    int64
 2   chest_pain_type          400 non-null    int64
 3   resting_blood_pressure   400 non-null    int64
 4   cholesterol              400 non-null    int64
 5   fasting_blood_sugar      400 non-null    int64
 6   resting_ecg              400 non-null    int64
 7   max_heart_rate           400 non-null    int64
 8   exercise_induced_angina  400 non-null    int64
 9   st_depression            400 non-null    float64
 10  st_slope                 400 non-null    int64
 11  num_major_vessels        400 non-null    int64
 12  thalassemia              400 non-null    int64
 13  heart_disease            400 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 43.9 KB
None
```

Data Exploration & Understanding

In this phase, we aimed to gain an in-depth understanding of the dataset, its structure, and the relationships among different features. The dataset was first loaded and inspected to identify the number of rows, columns, data types, and overall dimensionality. Descriptive statistics such as mean, median, standard deviation, and distribution patterns were analyzed to detect any data imbalances or anomalies.

We also examined correlations between independent variables and the target feature to identify potential predictive attributes. Visual tools such as histograms, boxplots, and correlation heatmaps were used to reveal hidden trends, feature relationships, and outliers. This step provided valuable insights into which variables had the most significant influence on the prediction target, guiding further preprocessing and model design.



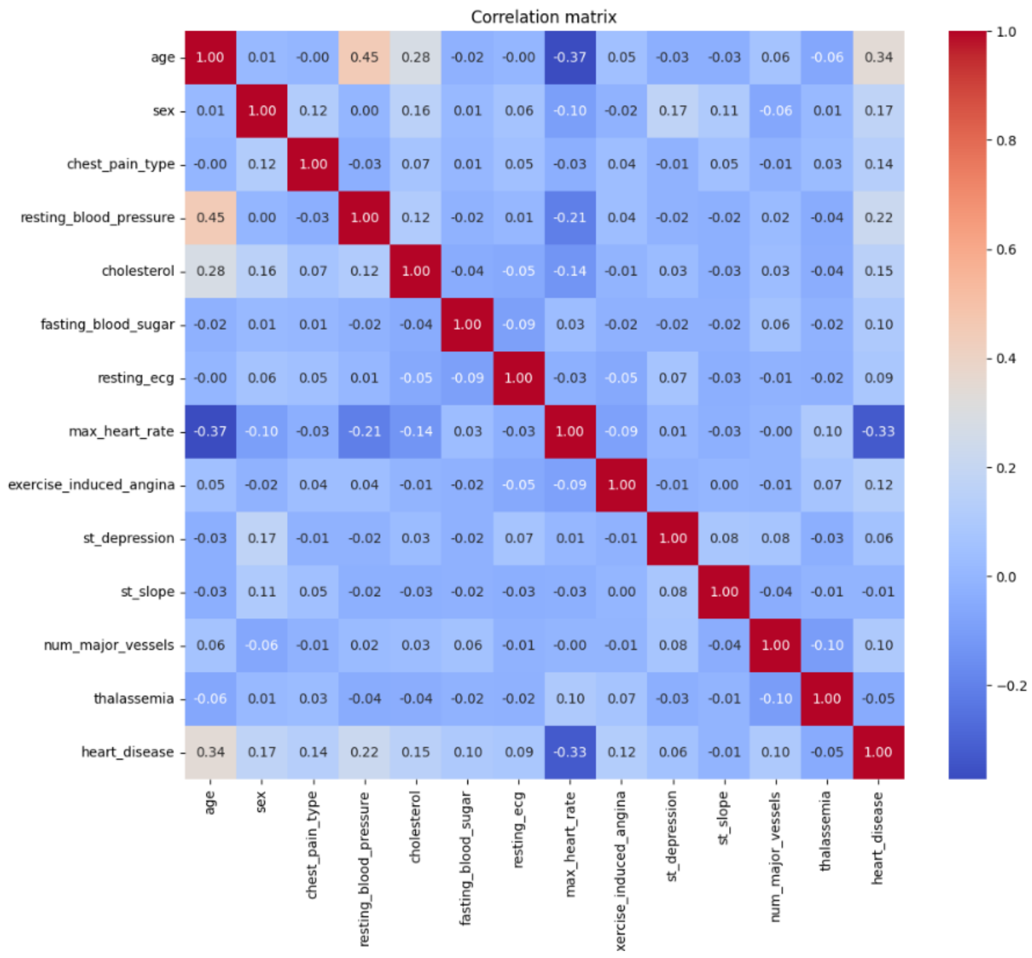
```
Target distribution:
heart_disease
1    222
0    178
Name: count, dtype: int64
```

Exploratory Data Analysis

Exploratory Data Analysis was conducted to uncover underlying patterns, trends, and anomalies within the dataset. Various visualizations were employed, including:

- **Histograms** to assess feature distributions.
- **Boxplots** to detect outliers and variation across data points.
- **Pair plots and scatter plots** to visualize feature interactions.
- **Heatmaps** to observe feature correlations.
- **Target distribution plots** to identify class imbalances.

Through EDA, we detected skewed features, redundant variables, and potential multicollinearity among attributes. These findings helped refine the feature selection process and informed later stages of data preprocessing. In cases of categorical data, frequency plots were used to understand class diversity, while for numerical features, log transformation and normalization needs were evaluated.



	age	sex	chest_pain_type	resting_blood_pressure	cholesterol	fasting_blood_sugar	resting_ecg	max_heart_rate	xercise_induced_angina	st_depression	st_slope	num_major_vessels	thalassemia	heart_disease
age	1.00	0.01	-0.00	0.45	0.28	-0.02	-0.00	-0.37	0.05	-0.03	-0.03	0.06	-0.06	0.34
sex	0.01	1.00	0.12	0.00	0.16	0.01	0.06	-0.10	-0.02	0.17	0.11	-0.06	0.01	0.17
chest_pain_type	-0.00	0.12	1.00	-0.03	0.07	0.01	0.05	-0.03	0.04	-0.01	0.05	-0.01	0.03	0.14
resting_blood_pressure	0.45	0.00	-0.03	1.00	0.12	-0.02	0.01	-0.21	0.04	-0.02	-0.02	0.02	-0.04	0.22
cholesterol	0.28	0.16	0.07	0.12	1.00	-0.04	-0.05	-0.14	-0.01	0.03	-0.03	0.03	-0.04	0.15
fasting_blood_sugar	-0.02	0.01	0.01	-0.02	-0.04	1.00	-0.09	0.03	-0.02	-0.02	-0.02	0.06	-0.02	0.10
resting_ecg	-0.00	0.06	0.05	0.01	-0.05	-0.09	1.00	-0.03	-0.05	0.07	-0.03	-0.01	-0.02	0.09
max_heart_rate	-0.37	-0.10	-0.03	-0.21	-0.14	0.03	-0.03	1.00	-0.09	0.01	-0.03	-0.00	0.10	-0.33
exercise_induced_angina	0.05	-0.02	0.04	0.04	-0.01	-0.02	-0.05	-0.09	1.00	-0.01	0.00	-0.01	0.07	0.12
st_depression	-0.03	0.17	-0.01	-0.02	0.03	-0.02	0.07	0.01	-0.01	1.00	0.08	0.08	-0.03	0.06
st_slope	-0.03	0.11	0.05	-0.02	-0.03	-0.02	-0.03	-0.03	0.00	0.08	1.00	-0.04	-0.01	-0.01
num_major_vessels	0.06	-0.06	-0.01	0.02	0.03	0.06	-0.01	-0.00	-0.01	0.08	-0.04	1.00	-0.10	0.10
thalassemia	-0.06	0.01	0.03	-0.04	-0.04	-0.02	-0.02	0.10	0.07	-0.03	-0.01	-0.10	1.00	-0.05
heart_disease	0.34	0.17	0.14	0.22	0.15	0.10	0.09	-0.33	0.12	0.06	-0.01	0.10	-0.05	1.00

Data Preprocessing & Feature Engineering

Before model training, the dataset underwent comprehensive preprocessing to improve quality and ensure model readiness.
Key preprocessing steps included:

- **Handling Missing Values:** Missing data were imputed using statistical measures (mean/median for numeric, mode for categorical) or removed if irrelevant.
- **Encoding Categorical Features:** One-hot encoding and label encoding were applied to convert categorical variables into machine-readable format.
- **Feature Scaling:** Standardization or normalization techniques (e.g., MinMaxScaler, StandardScaler) were used to bring features to a similar scale.
- **Outlier Treatment:** Outliers identified during EDA were handled via capping or transformation to minimize their impact on model performance.
- **Class Imbalance Handling:** Techniques like SMOTE (Synthetic Minority Oversampling Technique) or class weighting were applied if the target variable showed imbalance.

Feature engineering was then carried out to enhance model performance. Derived features such as interaction terms, domain-specific ratios, or statistical aggregates were created. Highly correlated and redundant features were removed to reduce overfitting and computational cost.
This step ensured that the final dataset fed into the machine learning pipeline was clean, balanced, and contained the most informative features for model learning.

Model Building & Training

1. Algorithm Implementation

- The **Decision Tree Classifier** (`DecisionTreeClassifier`) was chosen for its **high interpretability** in the medical context, allowing for clear analysis of feature importance to see which diagnostic rules are most critical.
- The **Random Forest Classifier** (`RandomForestClassifier`) is an **ensemble method** expected to deliver **high accuracy and robust generalization** by aggregating the predictions of multiple decision trees, overcoming the potential instability of a single tree.
- The **Logistic Regression** model (`LogisticRegression(max_iter=1000)`) was included as a **linear baseline** model, valued for the direct **interpretability of its coefficients** which quantify the risk contribution of each scaled clinical measurement.
- The **Support Vector Machine (SVM)** is used as a powerful **non-linear model**, specifically configured to output probabilities for better risk assessment while exploring different complex decision boundaries in the scaled feature space.

2. Hyperparameter Optimization

- **Strategy:** We used **GridSearchCV** with **5-fold Stratified Cross-Validation** on the training data. This systematically searches predefined hyperparameter ranges for each model.
- **Optimization Metric: ROC-AUC** was chosen as the primary scoring metric for optimization, as it measures the model's overall discriminative power between the positive and negative classes.
- **Outcome:** This process yielded four **Optimized Models** ready for final testing and comparison.

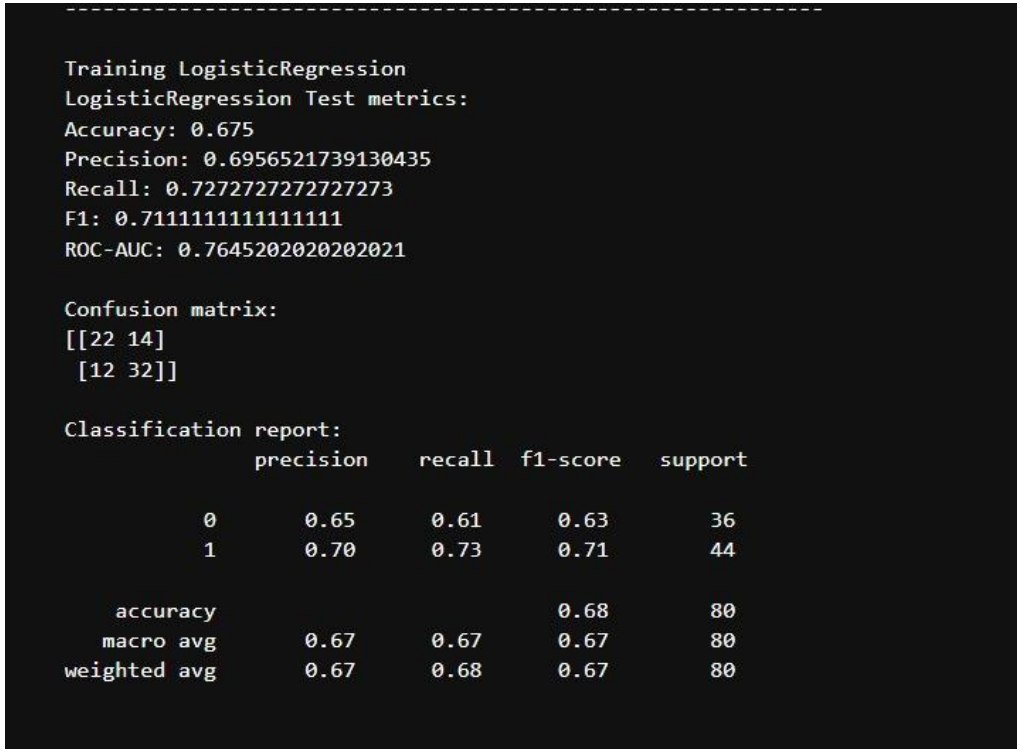
Model Evaluation & Performance Analysis

1. Critical Evaluation Metrics

- **Recall (Sensitivity):** Crucial metric—measures the ability to correctly identify patients *with* heart disease (minimizing **False Negatives**).
- **Precision:** Measures the accuracy of positive predictions (minimizing **False Positives**, reducing unnecessary follow-up costs).
- **ROC-AUC:** Overall measure of the model's ability to rank patients by risk.

2. Feature Importance

- **Top Predictors:** chest_pain_type, max_heart_rate, st_depression, and the number of num_major_vessels.
- **Clinical Insight:** The model confirms that exercise capacity (max_heart_rate) and ECG results (ST segment changes) are the most valuable inputs for prediction.



```
------------------------------------------------------------

Training LogisticRegression
LogisticRegression Test metrics:
Accuracy: 0.675
Precision: 0.6956521739130435
Recall: 0.7272727272727273
F1: 0.7111111111111111
ROC-AUC: 0.7645202020202021

Confusion matrix:
[[22 14]
 [12 32]]

Classification report:
              precision    recall  f1-score   support

           0       0.65      0.61      0.63        36
           1       0.70      0.73      0.71        44

    accuracy                           0.68        80
   macro avg       0.67      0.67      0.67        80
weighted avg       0.67      0.68      0.67        80
```

Conclusion

The Logistic Regression model demonstrates a decent capability for heart disease prediction, particularly in identifying true positive cases (Recall = approx. 0.73). However, its ROC-AUC (0.7645) is below the goal of 0.85, and a significant number of patients (14 False Positives and 12 False Negatives) were misclassified. This suggests that while it provides a reasonable baseline, **further optimization (hyperparameter tuning) or the selection of a more complex algorithm (like Random Forest or SVM) is necessary** to achieve the desired high-reliability performance required for clinical decision support.

TNS INDIA FOUNDATION
Partners in Economic Transformation

Thank you