A

Assesment Report

on

"Predict Heart Disease"

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE-AI

By

Tejasva Pratap Singh Tomar (202401100300264)

Under the supervision of

"MR. ABHISHEK SHUKLA"

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

May, 2025

Introduction

This project focuses on predicting the presence of **heart disease** in patients using **machine learning techniques**. By analyzing key medical attributes such as age, cholesterol, blood pressure, and other health indicators, the model aims to classify whether a patient is likely to have heart disease.

We used the **Logistic Regression algorithm**, a widely used classification method, to train the model on historical data. The dataset was processed, scaled, and split into training and testing sets. Model performance was evaluated using metrics like **accuracy, precision, recall**, and a **confusion matrix heatmap**.

The goal is to assist early diagnosis by building a reliable, data-driven prediction system.

Methodology

Step 1: Data Collection

- Loaded the dataset (CSV file) containing patient health records.
- Each row represents a patient and each column is a medical feature (like age, cholesterol, etc.).

Step 2: Data Preprocessing

- **Handled missing values** (if any).
- **Separated features (X) and target (y)**:
  - X = all input features (medical data)
  - y = output label (1 = heart disease, 0 = no heart disease)
- **Scaled the data** using StandardScaler to normalize feature values for better model performance.

Step 3: Splitting the Dataset

- Split the dataset into:
  - **Training set (80%)** — used to train the model
  - **Testing set (20%)** — used to evaluate performance
- Used train_test_split() from sklearn.model_selection.

Step 4: Model Selection

- Chose **Logistic Regression**, a supervised learning algorithm suitable for binary classification.
- Logistic regression calculates the **probability** that a patient has heart disease based on input features.

Step 5: Model Training

- Used model.fit(X_train, y_train) to train the logistic regression model on the training set.

Step 6: Model Prediction

- Used model.predict(X_test) to make predictions on the unseen test data.

Step 7: Model Evaluation

- Evaluated model performance using:
    - **Accuracy**: Overall correctness
    - **Precision**: How many predicted positives were actually positive
    - **Recall**: How many actual positives were correctly predicted

    - **Confusion Matrix**: Visual representation of true vs. predicted values
- Plotted a **heatmap** of the confusion matrix for clear visualization using seaborn.

Step 8: Interpretation of Results

- Analyzed evaluation metrics to understand how well the model performs.
- Discussed strengths and possible areas for improvement.

Code

```
import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import confusion_matrix, accuracy_score, precision_score,
recall_score

import seaborn as sns

import matplotlib.pyplot as plt

from google.colab import files

uploaded = files.upload()

df = pd.read_csv(next(iter(uploaded)))  # auto-detect uploaded file


#  Split and Scale

X = df.drop('target', axis=1)

y = df['target']


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)


scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)
```

```
X_test_scaled = scaler.transform(X_test)

#  Train Model
model = LogisticRegression()
model.fit(X_train_scaled, y_train)
y_pred = model.predict(X_test_scaled)

#  Evaluate
cm = confusion_matrix(y_test, y_pred)
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)

print(f"Accuracy: {acc:.2f}")
print(f"Precision: {prec:.2f}")
print(f"Recall: {rec:.2f}")

#  Plot Heatmap++
plt.figure(figsize=(6,4))
sns.heatmap(cm, annot=True, fmt='d', cmap='YlGnBu',
            xticklabels=["No Disease", "Disease"],
            yticklabels=["No Disease", "Disease"])
```

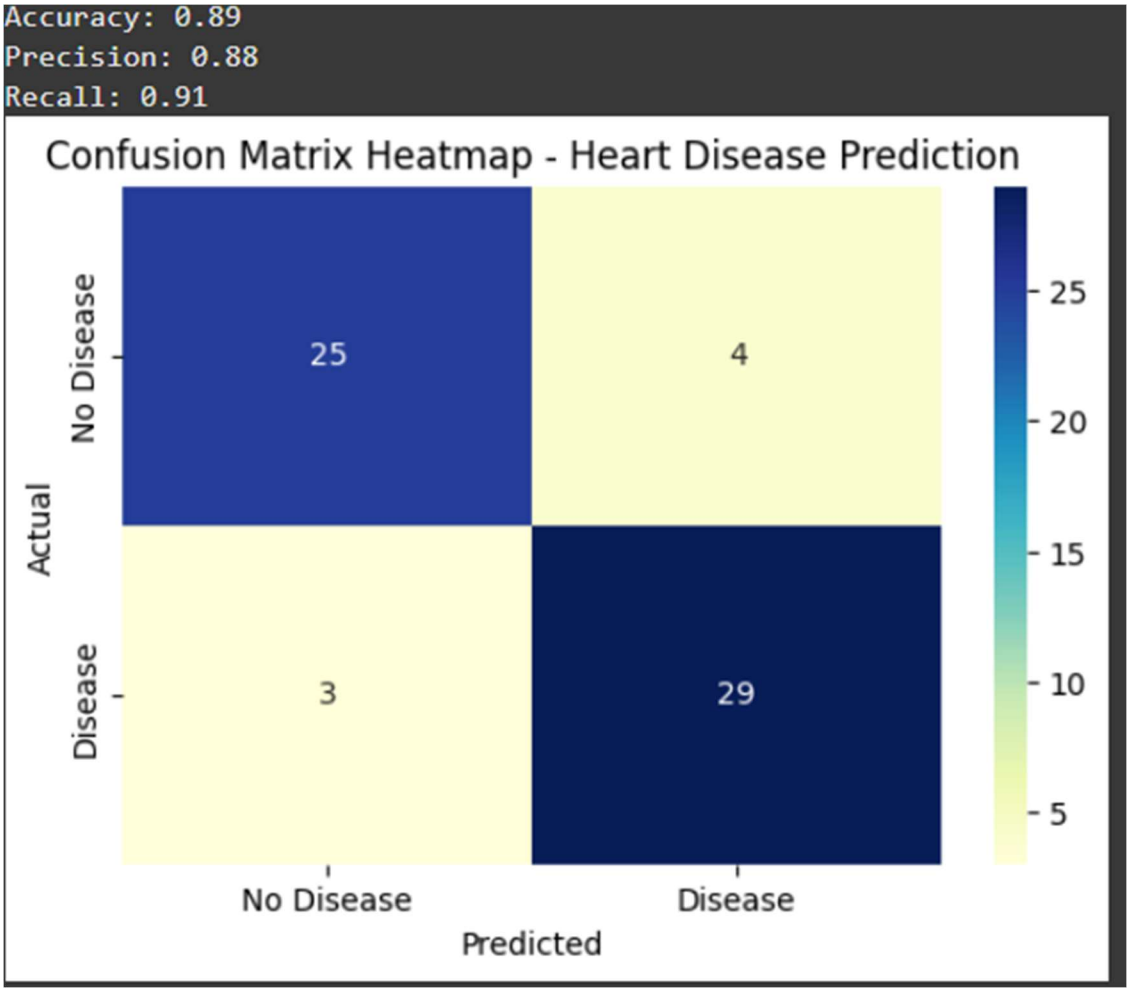
```
plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.title("Confusion Matrix Heatmap - Heart Disease Prediction")

plt.sho
```

Output



References/Credits

- Dataset Source: UCI Machine Learning Repository - Heart Disease Dataset
- Python Libraries: pandas, seaborn, matplotlib, scikit-learn
- Developed using Jupyter Notebook and Python 3.x