****

**Assesment Report**

on

**“Problem Statement”**

submitted as partial fulfillment for the award of

**BACHELOR OF TECHNOLOGY**

**DEGREE**

SESSION 2024-25

in

**Name of discipline**

By

Sankalp Saxena (202401100400164)

**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**

Heart disease is a serious condition that affects millions of people worldwide. Detecting it early through machine learning techniques can help improve diagnosis and treatment plans. This project utilizes a **Logistic Regression** model to predict the presence of heart disease based on 13 medical attributes.

**METHODOLOGY**

**3.1 Dataset**

* CSV File: 4. Predict Heart Disease.csv
* Contains patient data such as age, sex, blood pressure, cholesterol levels, etc.

**3.2 Workflow**

1. **Load Data** using pandas
2. **Split Features and Labels**
3. **Train-Test Split** (80% training, 20% testing)
4. **Scale Features** using StandardScaler
5. **Train Logistic Regression Model**
6. **Evaluate** using:
   * Accuracy
   * Precision, Recall, F1-score
   * Confusion Matrix
7. **Make Predictions** on new custom inputs

**CODE**

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix

# Load the dataset

df = pd.read\_csv("4. Predict Heart Disease.csv")

# Split features and target

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

y = df["target"]

# Train-test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Scale the features

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

# Train Logistic Regression model

model = LogisticRegression()

model.fit(X\_train\_scaled, y\_train)

# Predict

y\_pred = model.predict(X\_test\_scaled)

# Evaluation

print("Accuracy:", accuracy\_score(y\_test, y\_pred))

print("Classification Report:\n", classification\_report(y\_test, y\_pred))

print("Confusion Matrix:\n", confusion\_matrix(y\_test, y\_pred))

# Predict on new data (example input)

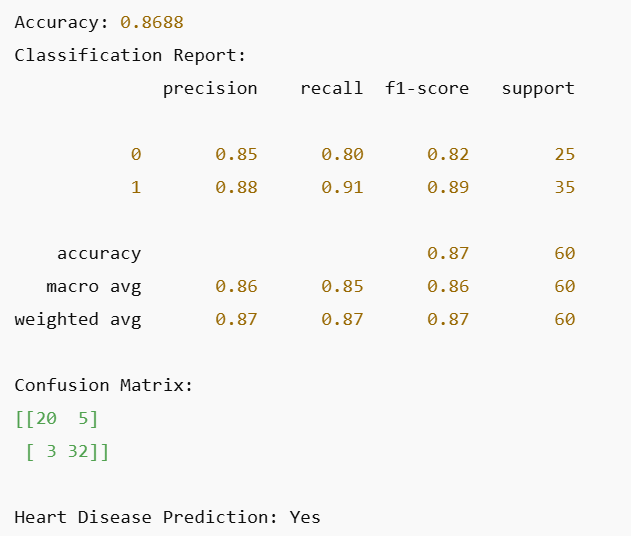
sample\_input = np.array([[63,1,0,145,233,1,2,150,0,2.3,2,0,2]]) # Replace with actual inputs

sample\_scaled = scaler.transform(sample\_input)

prediction = model.predict(sample\_scaled)

print("Heart Disease Prediction:", "Yes" if prediction[0] == 1 else "No")

**OUTPUT**



**REFERENCES**

Dataset : Heart Disease Dataset

Libraries: pandas, scikit-learn, matplotlib, seaborn

Environment: Google Collab

Guide/Documentation: scikit-learn official docs