****

**Assesment Report**

on

**“Predict Heart Disease”**

submitted as partial fulfillment for the award of

**BACHELOR OF TECHNOLOGY**

**DEGREE**

SESSION 2024-25

in

**CSE AIML**

By

Shubham Chaudhary

(202401100400184)

**Under the supervision of**

“Abhishek Shukla”

**KIET Group of Institutions, Ghaziabad**

**Title Page**

**Problem Statement: Predict the presence of heart disease using a given dataset and evaluate the performance of a classification model using metrics such as confusion matrix, accuracy, precision, and recall.**

**Submitted By: [Shubham Chaudhary]  
Roll No.: [202401100400184]  
Course: [Introduction to AI]  
Date: [18 April 2025]**

**Introduction**

**Heart disease is one of the leading causes of death globally. Early detection and diagnosis play a crucial role in prevention and treatment. In this project, we are using a heart disease dataset to predict the presence or absence of heart disease using machine learning classification techniques. This task involves loading the dataset, preprocessing the data, building a model, evaluating it using different performance metrics, and visualizing results using heatmaps.**

**The dataset used is a CSV file containing patient information and a target column indicating the presence (1) or absence (0) of heart disease.**

**Methodology**

1. **Data Loading: Imported the dataset using pandas.**
2. **Heart disease is one of the leading causes of death globally. Early detection and diagnosis play a crucial role in prevention and treatment. In this project, we are using a heart disease dataset to predict the presence or absence of heart disease using machine learning classification techniques. This task involves loading the dataset, preprocessing the data, evaluating it using different performance metrics, and visualizing results using heatmaps.**
3. **Preprocessing:**
   * **Encoded categorical features using get\_dummies().**
   * **Standardized features using StandardScaler().**
4. **Model Building:**
   * **Used RandomForestClassifier for classification.**
   * **Split the dataset into training and testing sets.**
5. **Model Evaluation:**
   * **Predicted the test results.**
   * **Evaluated the model using accuracy, precision, recall, and confusion matrix.**
6. **Visualization:**
   * **Plotted confusion matrix using seaborn heatmap.**
   * **Plotted feature importance to understand model interpretability.**

**CODE**

**import pandas as pd**

**import numpy as np**

**import seaborn as sns**

**import matplotlib.pyplot as plt**

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

**from sklearn.preprocessing import StandardScaler**

**from sklearn.ensemble import RandomForestClassifier**

**from sklearn.metrics import confusion\_matrix, classification\_report, accuracy\_score, precision\_score, recall\_score, f1\_score**

**# Load the dataset**

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

**# Encode categorical variables**

**df = pd.get\_dummies(df, drop\_first=True)**

**# Features and target**

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

**y = df["target"]**

**# Split data**

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

**# Scale features**

**scaler = StandardScaler()**

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

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

**# Train model**

**model = RandomForestClassifier(random\_state=42)**

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

**# Predict**

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

**# Confusion matrix**

**conf\_matrix = confusion\_matrix(y\_test, y\_pred)**

**plt.figure(figsize=(6,5))**

**sns.heatmap(conf\_matrix, annot=True, cmap='Blues', fmt='d')**

**plt.title("Confusion Matrix")**

**plt.xlabel("Predicted")**

**plt.ylabel("Actual")**

**plt.show()**

**# Metrics**

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

**print("Precision:", precision\_score(y\_test, y\_pred))**

**print("Recall:", recall\_score(y\_test, y\_pred))**

**print("F1 Score:", f1\_score(y\_test, y\_pred))**

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

**# Feature importances**

**importances = model.feature\_importances\_**

**feat\_names = X.columns**

**feat\_imp\_df = pd.DataFrame({'Feature': feat\_names, 'Importance': importances})**

**feat\_imp\_df = feat\_imp\_df.sort\_values(by='Importance', ascending=False)**

**plt.figure(figsize=(10, 6))**

**sns.barplot(x='Importance', y='Feature', data=feat\_imp\_df.head(10))**

**plt.title('Top 10 Important Features')**

**plt.tight\_layout()**

**plt.show()**

**Output/Result**

**Example outputs:**

* **Confusion Matrix heatmap image**
* **Top 10 Feature Importance bar chart**
* **Console output with accuracy, precision, recall, and F1-score**

**References/Credits**

* **Dataset: Predict Heart Disease.csv (provided in course material or available from UCI repository)**
* **Libraries used: pandas, numpy, seaborn, matplotlib, scikit-learn**
* **Visualizations inspired by Seaborn documentation**
* **RandomForestClassifier: Scikit-learn Documentation**