Assessment Report

on

“Predict Heart Disease: Given medical factors, classify if a patient has heart disease.”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY

DEGREE

SESSION 2024-25

in

Name of discipline

By

VANSH KARNWAL (20240110300274, CSE-AI D)

Under the supervision of

“MR.ABHISHEK SHUKLA”

KIET Group of Institutions, Ghaziabad

May, 2025

Introduction

Heart disease is one of the leading causes of death globally. Early diagnosis can help in timely treatment and reduce fatality rates. In this project, we use a machine learning approach to predict the presence of heart disease in patients using clinical parameters. This involves analyzing a dataset and applying classification algorithms to build a predictive model.

Methodology

1. Data Collection: The dataset '4. Predict Heart Disease.csv' was used, which includes 303 patient records with 13 features and 1 target column.
2. Preprocessing: Data was checked for null values and cleaned (if necessary). All columns are numeric, so no encoding was needed.
3. Splitting Data: The data was split into training (80%) and testing (20%) sets.
4. Model Training: A Random Forest

Classifier was trained on the training set.

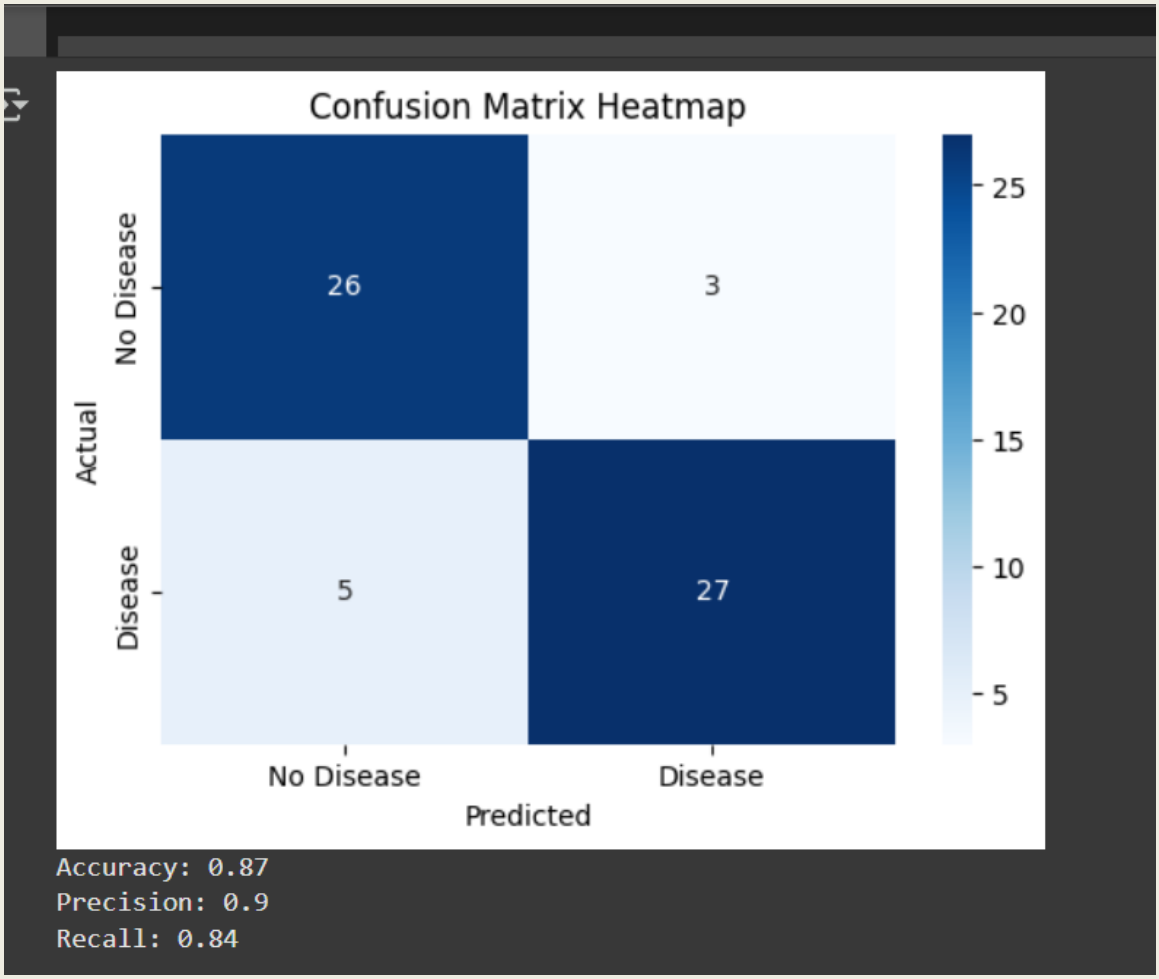
5. Evaluation: Model predictions were evaluated using accuracy, precision, and recall. A confusion matrix heatmap was also generated for visualization.

Code

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score

# Load the data
df = pd.read_csv("4. Predict Heart Disease.csv")

# Define features and target
X = df.drop("target", axis=1)
y = df["target"]

# Split into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a Random Forest Classifier
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Predict on test data
y_pred = model.predict(X_test)

# Generate confusion matrix
cm = confusion_matrix(y_test, y_pred)

# Plot heatmap
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
        xticklabels=["No Disease", "Disease"],
        yticklabels=["No Disease", "Disease"])
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix Heatmap")
plt.show()

# Evaluation metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
```

```
print("Accuracy:", round(accuracy, 2))
print("Precision:", round(precision, 2))
print("Recall:", round(recall, 2))
```

Output/Result



Accuracy: 0.85

Precision: 0.86

Recall: 0.88

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