Assessment Report

on

“ Predict Heart Disease”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY

DEGREE

SESSION 2024-25

in

Computer Science Engineering - Artificial Intelligence

(CSE-AI)

By

SUHAWANI SHUKLA (20240110300254, CSE-AI D)

Under the supervision of

“MR.ABHISHEK SHUKLA”

KIET Group of Institutions, Ghaziabad

May, 2025

Introduction

Heart disease is a leading cause of death worldwide, and early detection is key to improving patient outcomes. Traditional diagnostic methods can be time-consuming and costly, but machine learning provides an effective alternative. This project leverages machine learning to predict the likelihood of heart disease based on various health factors, such as age, cholesterol levels, and maximum heart rate. By training a logistic regression model on a dataset containing these features, the goal is to build a system that can predict whether a patient is at risk of heart disease.

METHODOLOGY

1.Data Collection:

The **Heart Disease UCI dataset** is used, containing medical features and a target variable indicating heart disease.

2. Data Preprocessing:

- **Label Encoding** for categorical variables (e.g., gender).
- **Feature Scaling** using **StandardScaler** to normalize numerical data.

2. **Model Selection**:
Logistic Regression is used for binary classification.

3. **Model Training and Evaluation**:
The model is trained on the training data, and performance is evaluated using metrics like

accuracy, **precision**, and **recall**.

Prediction:
 The model predicts heart disease for new patients, providing a classification and probability.

CODE

```
import pandas as pd
import numpy as np
from google.colab import files
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler, LabelEncoder
import sys

# Step 1: Upload CSV File (Colab or Jupyter)
def upload_csv():
    print("📂 Please upload your heart disease CSV file...")
    uploaded = files.upload()
    for fn in uploaded.keys():
        print(f"Successfully uploaded {fn}")
        return fn

# Step 2: Load Data and Automatically Extract Columns
def load_and_validate_data(path):
    df = pd.read_csv(path)
    print(f"\n✅ Loaded data with columns: {df.columns.tolist()}")

    # Checking if the target column exists
    if "target" not in df.columns:
        print("❌ Missing 'target' column in CSV. Exiting.")
        sys.exit()
    # Encode categorical columns (e.g., "sex") using LabelEncoder
    if 'sex' in df.columns:
        le = LabelEncoder()
        df['sex'] = le.fit_transform(df['sex'])  # Male -> 0, Female -> 1

    # Automatically use all columns except "target" as features
    X = df.drop("target", axis=1)
    y = df["target"]

    return df, X, y

# Step 3: Train the Model
def train_model(X, y):
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)
    model = LogisticRegression()
    model.fit(X_scaled, y)
    return model, scaler

# Step 4: Get Patient Data from User (Including Gender as 0 or 1)
def get_patient_data(columns):
    print("\n🩺 Enter patient details below:")
    values = []
    for col in columns:
        while True:
            try:
                # Special handling for 'sex' (gender)
                if col == 'sex':
                    gender = input(f"Gender (0 for Male, 1 for Female) for
{col}: ")
                    values.append(int(gender))
                else:
                    val = float(input(f"{col}: "))
                    values.append(val)
                break
            except ValueError:
                print("Invalid input. Please enter a numeric value.")
    return np.array(values).reshape(1, -1)

# Step 5: Predict and Display Result
```

```python
def predict_heart_disease(model, scaler, patient_data):
    scaled = scaler.transform(patient_data)
    prediction = model.predict(scaled)[0]
    probability = model.predict_proba(scaled)[0][prediction]
    print("\n📊 Prediction Result:")
    if prediction == 1:
        print(f"🔴 Patient is likely to have heart disease. (Confidence:
{probability:.2%})")
    else:
        print(f"🟢 Patient is unlikely to have heart disease. (Confidence:
{probability:.2%})")

# ---- Main Flow ----
if __name__ == "__main__":
    # Step 1: Upload CSV file
    file_path = upload_csv()

    # Step 2: Load and validate the data from the uploaded CSV
    df, X, y = load_and_validate_data(file_path)

    # Step 3: Train the Logistic Regression model
    model, scaler = train_model(X, y)

    # Step 4: Get new patient data from user input
    patient_input = get_patient_data(X.columns)

    # Step 5: Predict and display the result
    predict_heart_disease(model, scaler, patient_input)
```

Output/Result

```
Loaded data with columns: ['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs',
'restecg', 'thalach', 'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target']

🩺 Enter patient details below:
```

```
age: 45
Gender (0 for Male, 1 for Female) for sex: 1
cp: 76
trestbps: 87
chol: 56
fbs: 476
restecg: 687
thalach: 456
exang: 67
oldpeak: 87
slope: 468
ca: 45
thal: 636

📊 Prediction Result:
🔴 Patient is likely to have heart disease. (Confidence: 100.00%)
```

References/Credits

Dataset:

- **Heart Disease UCI dataset**: UCI Repository

Libraries & Algorithms:

- **Logistic Regression**: Scikit-learn
- **Pandas**: Pandas Documentation
- **Scikit-learn**: Scikit-learn Docs
- **Tkinter**: Tkinter Documentation