submitted as partial

fulfillment for the award of

BACHELOR OF TECHNOLOGY

DEGREE

SESSION 2024-25

in

CSE(Artificial intelligence and machine learning)-‘C’

By

Ravi Kishan (202401100400154)

Under the supervision of

“Abhishek Shukla”

KIET Group of Institutions, Ghaziabad

Affiliated to

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

(Formerly UPTU)

April, 2025

Methodology:-

1. Data Collection

The dataset used is the **Heart Disease Dataset** from the **UCI Machine Learning Repository**. It contains records of patients with and without heart disease, described by 13 clinical attributes.

2. Data Preprocessing

- **Handling Missing Values:** Impute or drop missing entries.
- **Feature Encoding:** Convert categorical data (e.g., chest pain type, thalassemia) using one-hot encoding or label encoding.
- **Scaling:** Normalize numeric features to bring them to the same scale.
- **Splitting Dataset:** Divide the data into training (80%) and testing (20%) sets.

3. Model Selection

The **Random Forest Classifier** is chosen for its robustness and accuracy. It reduces overfitting and works well with both categorical and numerical features

4. Model Evaluation

Evaluation metrics include:

- **Accuracy**
- **Precision**
- **Recall**
- **F1-score**
- **Confusion Matrix**
- **ROC-AUC Curve**

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")
```

Result:-

```
Accuracy: 0.8852459016393442
Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.86      0.88        29
           1       0.88      0.91      0.89        32

    accuracy                           0.89        61
   macro avg       0.89      0.88      0.88        61
weighted avg       0.89      0.89      0.89        61

Confusion Matrix:
 [[25  4]
 [ 3 29]]
Heart Disease Prediction: No
/usr/local/lib/python3.11/dist-packages/sklearn/utils/validation.py:27
  warnings.warn(
```

Refrences:-

Title: predict heart disease Dataset

Source: UCI Machine Learning Repository

Original Donor: Hungarian Institute of Cardiology, Budapest; University Hospital, Zurich; V.A. Medical Center, Long Beach, and Cleveland Clinic Foundation
Link: https://archive.ics.uci.edu/ml/datasets/heart+Disease
License: Public domain / Open Data
Citation:

Detrano, R., Janosi, A., Steinbrunn, W., Pfisterer, M., Schmid, J.J., Sandhu, S., Guppy, K., Lee, S., & Froelicher, V. (1989). ***International application of a new probability algorithm for the diagnosis of coronary artery disease*****. The American Journal of Cardiology, 64(5), 304–310.**