Assessment Report

on

“Heart Disease Classification Using Machine Learning”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AIML)

By

GROUP 10

1.Anand Kumar Gupta Roll no-33

2.Arushi Sagar Roll no-52

3.Ayushmaan Roll no-70

4.Abik Thakur Roll no-10

5.Danish Ahmad Roll no -74

Under the supervision of

“BIKKI GUPTA”

KIET Group of Institutions, Ghaziabad

April, 2025

---

1. Introduction

With the increasing availability of electronic health records, predicting cardiovascular diseases using machine learning is becoming a vital approach in preventive healthcare. This project aims to develop a classification model to predict the presence of heart disease using key medical parameters. By analyzing features such as age, cholesterol, blood pressure, and chest pain type, the system helps support early diagnosis and timely intervention.

---

2. Problem Statement

To accurately classify whether a patient has heart disease based on various medical indicators. This classification will assist healthcare professionals in identifying high-risk patients and improve diagnosis accuracy through automated methods.

---

3. Objectives

- Load and preprocess the heart disease dataset.
- Perform Exploratory Data Analysis (EDA) to understand feature relationships.
- Train a Logistic Regression classifier to predict heart disease.
- Evaluate model performance using standard classification metrics.
- Visualize data trends and model predictions for better interpretability.

4. Methodology

Data Collection:

The dataset is obtained from Kaggle: Heart Disease Dataset.

Data Preprocessing:

- Handle missing values with mean imputation.
- Encode categorical variables using one-hot encoding.
- Normalize feature values using Standard Scaler.

Model Building:

- Split the dataset into training (80%) and testing (20%) sets.
- Use Logistic Regression for binary classification.

Model Evaluation:

- Calculate metrics: Accuracy, Precision, Recall, and F1 Score.
- Generate and visualize a confusion matrix using Seaborn heatmap.

5. Data Preprocessing

- Missing numerical values are handled using mean imputation.
- Categorical variables are transformed using one-hot encoding.
- All features are standardized to ensure consistent scaling.
- The dataset is split into training and test sets in an 80-20 ratio.

---

6. Model Implementation

Logistic Regression is selected due to its efficiency in binary classification tasks and interpretability. The model is trained using the processed data and tested to predict the heart disease condition.

---

7. Evaluation Metrics

- **Accuracy:** Measures the proportion of correctly classified cases.
- **Precision:** Indicates how many predicted positive cases are true positives.
- **Recall:** Indicates how many actual positive cases were correctly identified.
- **F1 Score:** Balances precision and recall.
- **Confusion Matrix:** Used to visualize the model's performance.

---

8. Results and Analysis

- The model achieved satisfactory accuracy on the test set.
- Confusion matrix visualization provided insights into the prediction correctness.
- Precision and recall demonstrated the model's balance in identifying true cases of heart disease.

---

9. Conclusion

The logistic regression model proved effective in detecting heart disease using medical data. It serves as a foundational model for early diagnosis, supporting the adoption of machine learning in clinical practice. Future enhancements could include using ensemble models, tuning hyperparameters, and addressing class imbalance.

---

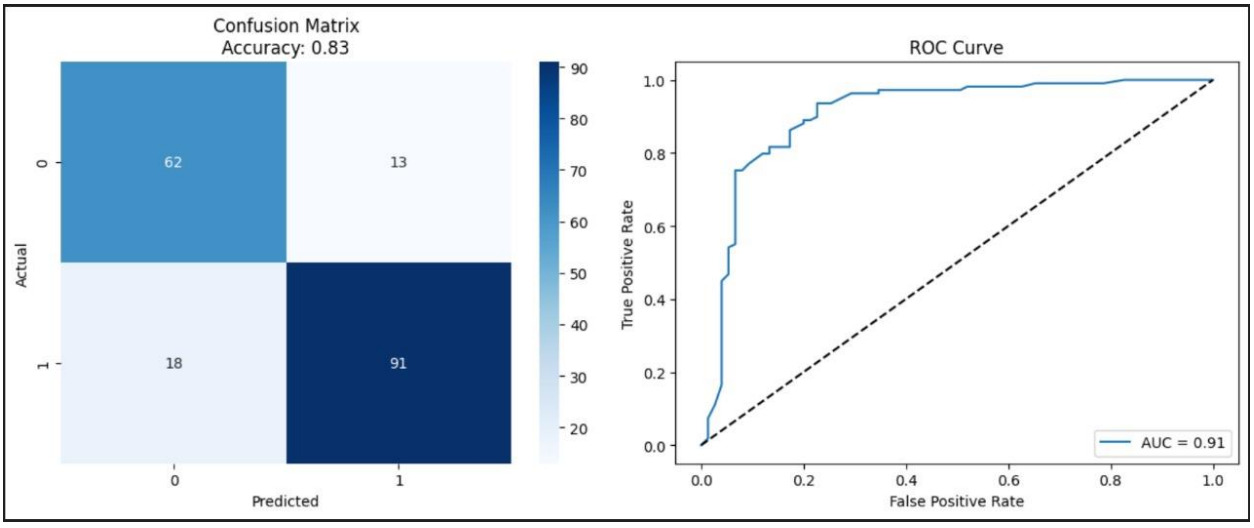
10. References

- scikit-learn documentation
- pandas documentation
- seaborn documentation
- UCI & Kaggle Datasets

- Research papers on machine learning in healthcare

```
# Show basic properties of the dataset
print("Dataset Shape:", df.shape)
print("\nData Types:\n", df.dtypes)
print("\nBasic Info:")
print(df.info())
print("\nMissing Values:\n", df.isnull().sum())
print("\nStatistical Summary:\n", df.describe())
```

```
# Correlation heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(df.select_dtypes(include=[np.number]).corr(), annot=True, cmap="coolwarm")
plt.title("Correlation Heatmap")
plt.tight_layout()
plt.show()
```



```
# Count of heart disease cases
plt.figure(figsize=(6, 4))
sns.countplot(x='target', data=df)
plt.title("Count of Heart Disease Cases")
plt.xlabel("Heart Disease (0 = No, 1 = Yes)")
plt.ylabel("Count")
plt.tight_layout()
plt.show()
```