Assessment Report

on

"Predict Heart Disease"

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

Name of discipline

By

Deepanshu Bhardwaj

202401100400077

CSE(Ai&Ml)- B

Under the supervision of

"Mr.Abhishek shukla"

KIET Group of Institutions, Ghaziabad

May, 2025

Introduction

Heart disease remains one of the leading causes of death globally. Early detection can drastically improve the treatment process and save lives. With the increasing amount of healthcare data, machine learning (ML) models have become a powerful tool in predicting the likelihood of heart disease. The purpose of this project is to build a predictive model using machine learning algorithms to identify patients at risk for heart disease, based on various health indicators such as age, gender, blood pressure, cholesterol levels, etc.

Methodology

The dataset used for this project is the **UCI Heart Disease dataset**, which contains 303 instances and 14 attributes including:

- Age
- Sex
- Chest pain type
- Resting blood pressure
- Serum cholesterol
- Fasting blood sugar
- Resting electrocardiographic results
- Maximum heart rate achieved
- Exercise induced angina
- Oldpeak (depression induced by exercise relative to rest)
- Slope of the peak exercise ST segment
- Number of major vessels colored by fluoroscopy
- Thalassemia

Steps Taken:

1. **Data Preprocessing**:
    - Handling missing values using imputation.
    - Feature normalization to ensure all features are on the same scale.

- Encoding categorical variables such as gender and chest pain type.
- Splitting the dataset into training and testing sets.

2. **Model Selection**: The following models were considered and implemented:
    - **Logistic Regression**: A baseline model to evaluate the relationship between input features and the outcome.
    - **Random Forest Classifier**: An ensemble model that combines multiple decision trees to improve performance.
    - **Support Vector Machine (SVM)**: A powerful classifier that works well with complex datasets.
3. **Evaluation Metrics**: The models were evaluated using common classification metrics:
    - **Accuracy**: The proportion of correct predictions.
    - **Precision**: The ability of the model to identify positive instances correctly.
    - **Recall**: The ability of the model to find all the positive instances.
    - **F1-score**: A balance between precision and recall.

Code

```
# Step 1: Import necessary libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns


from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import confusion_matrix, accuracy_score, classification_report, precision_score, recall_score


# Step 2: Load the data

df = pd.read_csv("/content/4. Predict Heart Disease.csv")  # Ensure this file is uploaded in Colab

print("Sample data:", df.head())


# Step 3: Explore the data

print("Dataset Shape:", df.shape)

df.info()

print("Summary:", df.describe())


# Step 4: Missing values
```

```
print("Missing values:", df.isnull().sum())


# Step 5: Target distribution visualization
sns.countplot(x='target', data=df, palette='Set2')
plt.title("Heart Disease Distribution (1 = Yes, 0 = No)")
plt.xlabel("Target")
plt.ylabel("Count")
plt.grid(True)
plt.show()


# Step 6: Feature and target selection
X = df.drop("target", axis=1)
y = df["target"]


# Step 7: Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


# Step 8: Feature scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)


# Step 9: Model training
model = RandomForestClassifier(random_state=42)
model.fit(X_train_scaled, y_train)
```

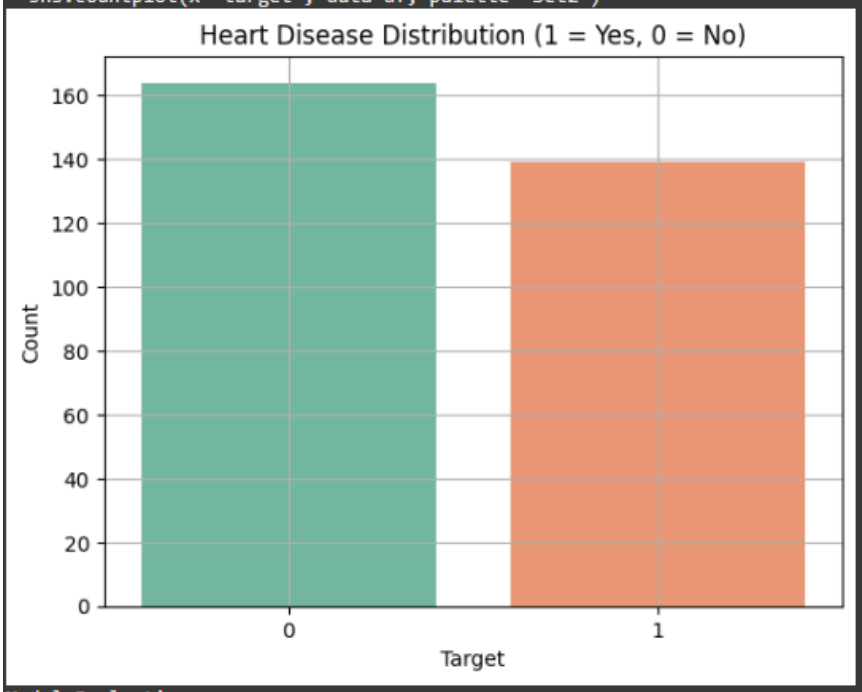
```
# Step 10: Make predictions
y_pred = model.predict(X_test_scaled)

# Step 11: Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
report = classification_report(y_test, y_pred)

print("Model Evaluation:")
print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("Classification Report:", report)

# Step 12: Show confusion matrix as heatmap
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues")
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

OUTPUT

```
Sample data:    age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  \
0   63    1   0       145   233    1        2      150      0      2.3      2
1   67    1   3       160   286    0        2      108      1      1.5      1
2   67    1   3       120   229    0        2      129      1      2.6      1
3   37    1   2       130   250    0        0      187      0      3.5      2
4   41    0   1       130   204    0        2      172      0      1.4      0

   ca  thal  target
0   0     2       0
1   3     1       1
2   2     3       1
3   0     1       0
4   0     1       0
Dataset Shape: (303, 14)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trestbps  303 non-null    int64
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
 7   thalach   303 non-null    int64
 8   exang     303 non-null    int64
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64
 11  ca        303 non-null    int64
 12  thal      303 non-null    int64
 13  target    303 non-null    int64
dtypes: float64(1), int64(13)
```

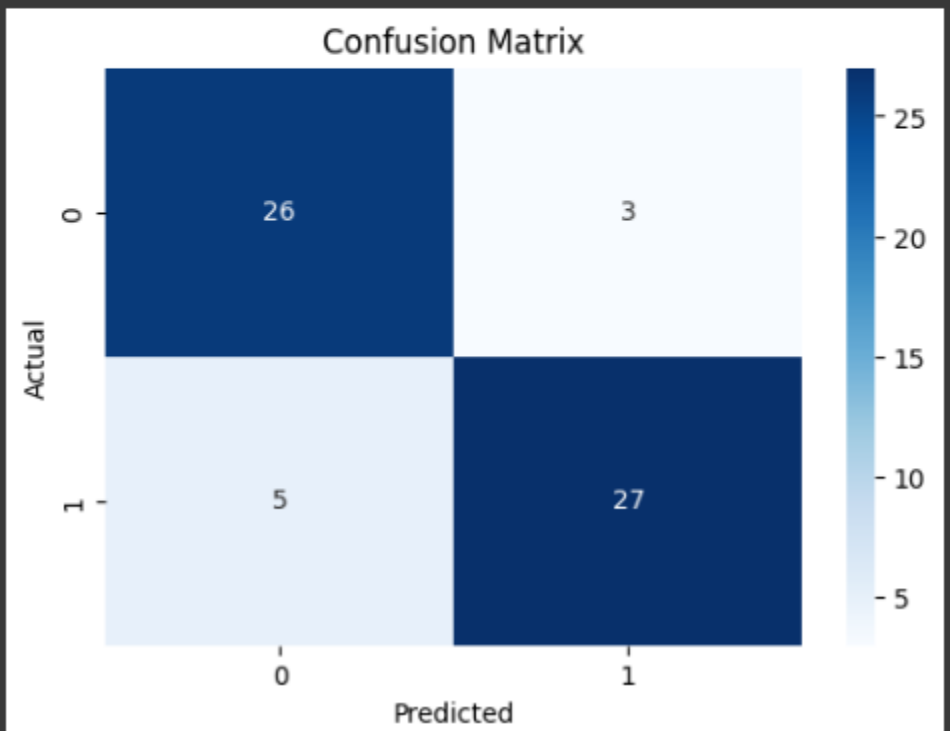
```
memory usage: 33.3 KB
Summary:              age         sex          cp    trestbps        chol         fbs  \
count  303.000000  303.000000  303.000000  303.000000  303.000000  303.000000
mean    54.438944    0.679868    2.158416  131.689769  246.693069    0.148515
std      9.038662    0.467299    0.960126   17.599748   51.776918    0.356198
min     29.000000    0.000000    0.000000   94.000000  126.000000    0.000000
25%     48.000000    0.000000    2.000000  120.000000  211.000000    0.000000
50%     56.000000    1.000000    2.000000  130.000000  241.000000    0.000000
75%     61.000000    1.000000    3.000000  140.000000  275.000000    0.000000
max     77.000000    1.000000    3.000000  200.000000  564.000000    1.000000

          restecg     thalach       exang     oldpeak       slope          ca  \
count  303.000000  303.000000  303.000000  303.000000  303.000000  303.000000
mean     0.990099  149.607261    0.326733    1.039604    0.600660    0.663366
std      0.994971   22.875003    0.469794    1.161075    0.616226    0.934375
min      0.000000   71.000000    0.000000    0.000000    0.000000    0.000000
25%      0.000000  133.500000    0.000000    0.000000    0.000000    0.000000
50%      1.000000  153.000000    0.000000    0.800000    1.000000    0.000000
75%      2.000000  166.000000    1.000000    1.600000    1.000000    1.000000
max      2.000000  202.000000    1.000000    6.200000    2.000000    3.000000

             thal      target
count  303.000000  303.000000
mean     1.831683    0.458746
std      0.956705    0.499120
min      1.000000    0.000000
25%      1.000000    0.000000
50%      1.000000    0.000000
75%      3.000000    1.000000
max      3.000000    1.000000
Missing values: age         0
sex         0
cp          0
trestbps    0
```

```
target      0
dtype: int64
<ipython-input-11-c785f5026c46>:27: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variab

  sns.countplot(x='target', data=df, palette='Set2')
```




```
Model Evaluation:
Accuracy: 0.8688524590163934
Precision: 0.9
Recall: 0.84375
Classification Report:              precision    recall  f1-score   support

           0       0.84      0.90      0.87        29
           1       0.90      0.84      0.87        32

    accuracy                           0.87        61
   macro avg       0.87      0.87      0.87        61
weighted avg       0.87      0.87      0.87        61
```

```
Recall: 0.84375
Classification Report:              precision    recall  f1-score   support

           0       0.84      0.90      0.87        29
           1       0.90      0.84      0.87        32

    accuracy                           0.87        61
   macro avg       0.87      0.87      0.87        61
weighted avg       0.87      0.87      0.87        61
```



References

- UCI Machine Learning Repository: Heart Disease Dataset.
- Tools & libraries: pyhton,pandas,matplotlib,seaborn.
- Machine Learning for Healthcare
- Documentation of the machine learning models used: Logistic Regression, Random Forest, SVM.
- Took help of chat gpt.