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

NIKHIL KUMAR (202401100400127)

Under the supervision of

Abhishek Shulka

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow
(Formerly UPTU)
May, 2025

Heart Disease Prediction Using Machine Learning

INTRODUCTION

Heart disease remains a leading cause of mortality worldwide, accounting for approximately 17.9 million deaths annually. Early detection is crucial for effective treatment and reducing the risk of severe complications. Traditional diagnostic methods often rely on subjective assessments and may not capture subtle patterns indicative of heart conditions.

In this project, we aim to develop a machine learning model that predicts the presence of heart disease based on various medical parameters. By analyzing features such as age, sex, blood pressure, cholesterol levels, and other relevant health indicators, the model can assist healthcare professionals in identifying high-risk individuals. This approach leverages data-driven insights to enhance diagnostic accuracy and support timely medical interventions.

Implementing such predictive models can be particularly beneficial in resource-constrained settings, enabling proactive healthcare measures and potentially reducing the burden on medical facilities.

Heart disease is a leading cause of death globally, often progressing silently until severe complications arise. Early detection is crucial for effective treatment and improved patient outcomes. Traditional diagnostic methods can be time-consuming and may not always detect the disease in its early stages. With the increasing availability of patient health data, machine learning (ML) offers a promising approach to enhance early diagnosis.

This project aims to develop a machine learning model that predicts the presence of heart disease based on various medical parameters such as age, blood pressure, cholesterol levels, and other relevant health indicators. By training the model on historical patient data, it can learn patterns associated with heart

disease and assist healthcare professionals in identifying high-risk individuals. Implementing such predictive models can lead to timely interventions, personalized treatment plans, and potentially reduce the burden on healthcare systems.

The integration of machine learning into heart disease prediction aligns with the broader movement towards data-driven healthcare solutions. Studies have demonstrated the effectiveness of ML algorithms in accurately predicting heart disease, highlighting their potential to transform traditional diagnostic practices . By leveraging these technologies, we can move towards more proactive and preventive healthcare strategies

Approach to Heart Disease Prediction

1. Data Collection and Understanding

We begin by gathering a dataset containing patient health records. This dataset includes features such as:

- Age
- Sex
- Chest pain type
- Resting blood pressure
- Serum cholesterol
- Fasting blood sugar
- Resting electrocardiographic results
- Maximum heart rate achieved

- Exercise-induced angina
- Oldpeak (ST depression induced by exercise)
- Slope of the peak exercise ST segment
- Number of major vessels colored by fluoroscopy
- Thalassemia
- Target variable indicating the presence (1) or absence (0) of heart disease Nature+7jmai.amegroups.org+7GeeksforGeeks+7arXiv+1jmai.amegroups.org+1

Understanding the dataset's structure and the significance of each feature is crucial for effective preprocessing and model selection.

2. Data Preprocessing

Data preprocessing ensures the dataset is clean and suitable for modeling:

- **Handling Missing Values**: Identify and address any missing or null values in the dataset.
- **Encoding Categorical Variables**: Convert categorical features into numerical values using techniques like one-hot encoding.
- **Feature Scaling**: Standardize numerical features to ensure they contribute equally to the model's learning process.

3. Exploratory Data Analysis (EDA)

EDA helps in understanding the relationships between features and the target variable:

- **Correlation Matrix**: Visualize correlations between features to identify multicollinearity.
- **Distribution Plots**: Analyze the distribution of features to detect skewness or outliers.

- **Class Balance**: Check the balance between classes in the target variable to determine if resampling techniques are needed.

4. Feature Selection

Selecting the most relevant features enhances model performance:

- **Statistical Tests**: Apply tests like Chi-square to assess the significance of features.
- **Model-Based Selection**: Use algorithms like Random Forest to determine feature importance.

5. Model Selection and Training

Choose appropriate machine learning algorithms based on the problem and dataset characteristics:

- **Logistic Regression**: Suitable for binary classification problems.
- **Decision Trees and Random Forest**: Handle non-linear relationships and interactions between features.
- **Support Vector Machines (SVM)**: Effective in high-dimensional spaces.
- **K-Nearest Neighbors (KNN)**: Simple and effective for small datasets.
- **Neural Networks**: Capture complex patterns in data.

Train the selected model(s) using the preprocessed dataset.

6. Model Evaluation

Assess the model's performance using appropriate metrics:

- **Accuracy**: Overall correctness of the model.
- **Precision and Recall**: Evaluate the model's ability to correctly identify positive cases.
- **F1-Score**: Harmonic mean of precision and recall.

- **Confusion Matrix**: Detailed breakdown of true positives, true negatives, false positives, and false negatives.
- **ROC-AUC Curve**: Measure the model's ability to distinguish between classes.

7. Hyperparameter Tuning

Optimize model performance by tuning hyperparameters using techniques like Grid Search or Random Search.

8. Model Deployment

Once a satisfactory model is developed, deploy it for practical use:

- **Web Application**: Integrate the model into a user-friendly interface using frameworks like Flask or Streamlit.
- **API Development**: Create APIs to allow other applications to access the model's predictions.

9. Monitoring and Maintenance

Continuously monitor the model's performance in the real world and update it as necessary to maintain accuracy and relevance.

CODE :

```
# 1. Upload and Read the CSV

from google.colab import files

uploaded = files.upload()


import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt


# Load dataset

df = pd.read_csv('4. Predict Heart Disease.csv')


# 2. Basic Info

print("Shape of dataset:", df.shape)

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

print(df.head())


# 3. Data Preprocessing

target_col = 'target' if 'target' in df.columns else df.columns[-1]


X = df.drop(target_col, axis=1)

y = df[target_col]
```

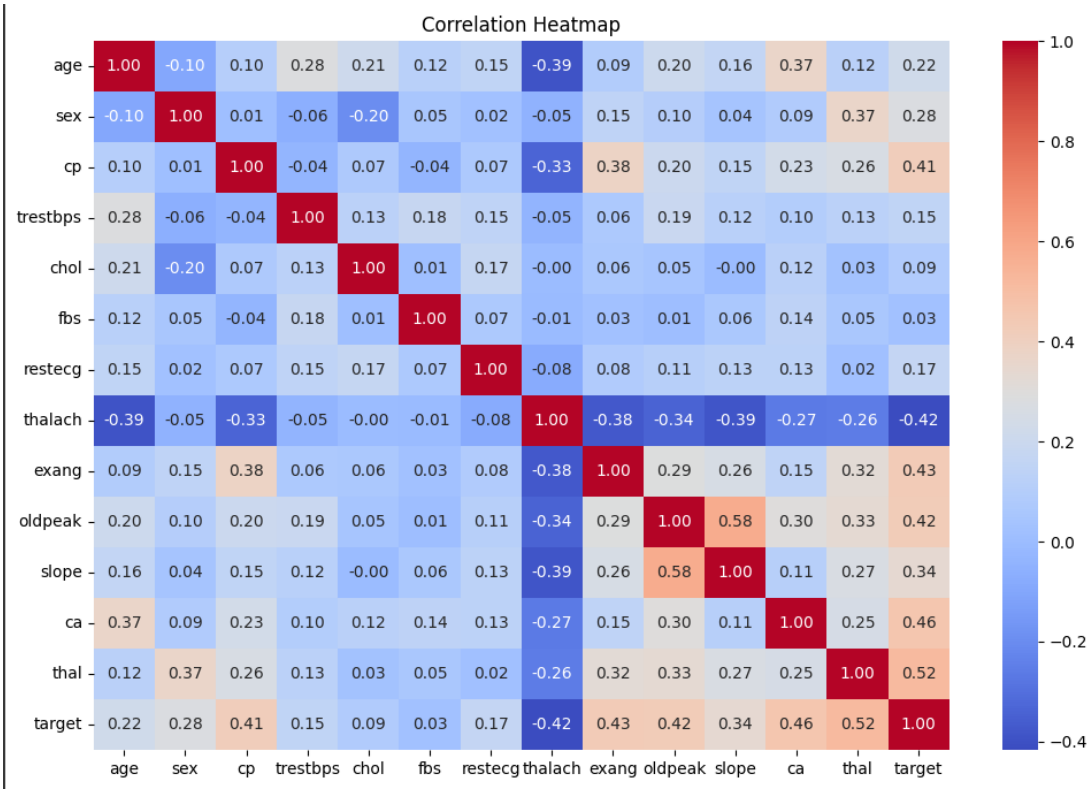
```
# Encode categorical variables if any
categorical_cols = X.select_dtypes(include=['object']).columns
if len(categorical_cols) > 0:
    X = pd.get_dummies(X, drop_first=True)

# Feature Scaling
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# 4. Train/Test Split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# 5. Train the Model
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# 6. Evaluation
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

y_pred = model.predict(X_test)
```

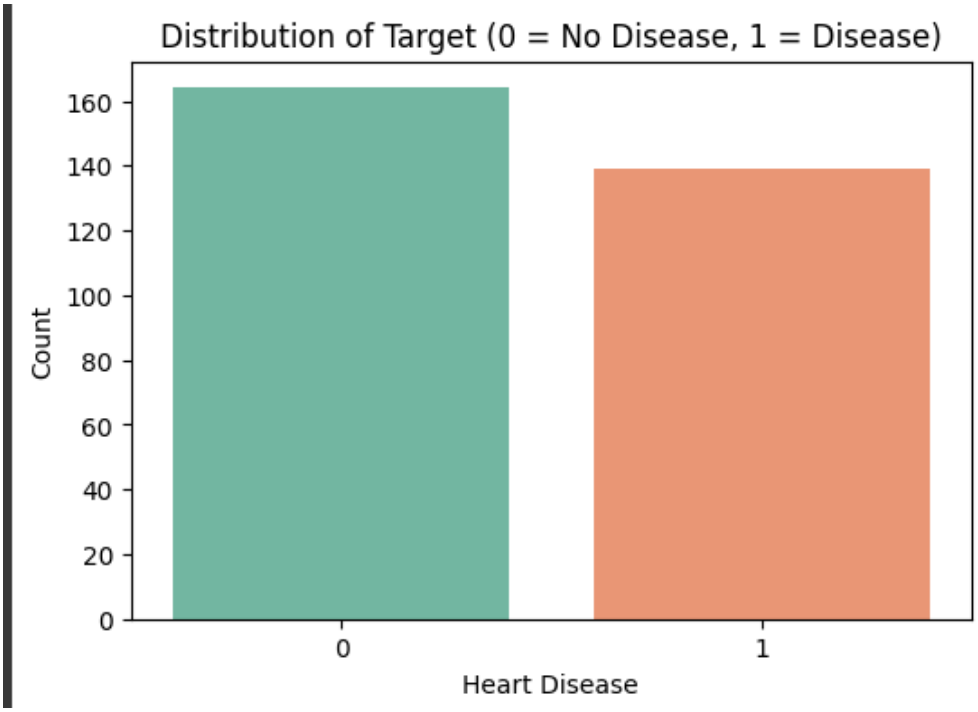
```
print("Accuracy Score:", accuracy_score(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))

# ==========================
# 7. PLOTS / GRAPHS
# ==========================

# Correlation Heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(df.corr(), annot=True, fmt=".2f", cmap="coolwarm")
plt.title("Correlation Heatmap")
plt.show()

# Countplot for Target Variable
plt.figure(figsize=(6, 4))
sns.countplot(x=target_col, data=df, palette='Set2')
plt.title("Distribution of Target (0 = No Disease, 1 = Disease)")
plt.xlabel("Heart Disease")
plt.ylabel("Count")
plt.show()

# Feature Importance Plot
importances = model.feature_importances_
```

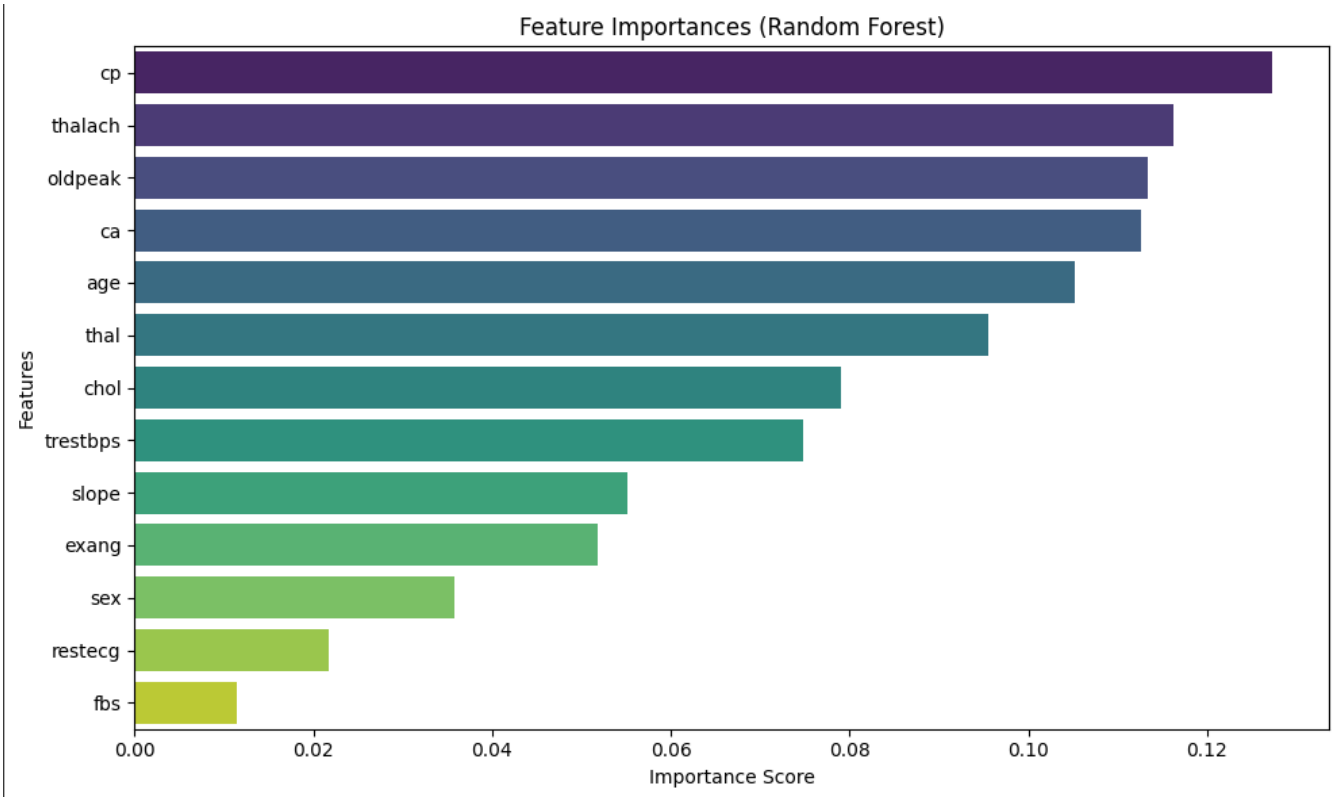
```
feature_names = X.columns


feat_imp_df = pd.DataFrame({'Feature': feature_names, 'Importance': importances})
feat_imp_df = feat_imp_df.sort_values(by='Importance', ascending=False)


plt.figure(figsize=(10, 6))
sns.barplot(x='Importance', y='Feature', data=feat_imp_df, palette='viridis')
plt.title("Feature Importances (Random Forest)")
plt.xlabel("Importance Score")
plt.ylabel("Features")
plt.tight_layout()
plt.show()
```

OUTPUT:

```
   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
Accuracy Score: 0.8688524590163934
Confusion Matrix:
 [[26  3]
 [ 5 27]]
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
```




Distribution of Target (0 = No Disease, 1 = Disease)
Count
160
140
120
100
80
60
40
20
0
0
1
Heart Disease


Feature Importances (Random Forest)
Features
cp
thalach
oldpeak
ca
age
thal
chol
trestbps
slope
exang
sex
restecg
fbs
0.00
0.02
0.04
0.06
0.08
0.10
0.12
Importance Score