

HEART DISEASE PREDICTION

MINI PROJECT REPORT

Submitted By

SHYAM PRASATH R - 211701052

VENKATESH C - 211701059

In partial fulfilment for the award of the degree

Of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND DESIGN



RAJALAKSHMI
ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai



RAJALAKSHMI ENGINEERING COLLEGE,

ANNA UNIVERSITY, CHENNAI - 600 025

NOVEMBER 2024

RAJALAKSHMI ENGINEERING COLLEGE

CHENNAI – 602105

BONAFIDE CERTIFICATE

Certified that this Report titled “**HEART DISEASE PREDICTION**” is the Bonafide work of “**Shyam Prasath R (211701052) & Venkatesh C (211701059)**” who carried out the work under my supervision.

Prof. Uma Maheshwar Rao,
Professor and Head
Department of
Computer Science and Design
Rajalakshmi Engineering College
Rajalakshmi Nagar
Thandalam
Chennai - 602105

Dr.P. Revathy, M.E., Ph.D.,
SUPERVISOR
Professor
Department of
Computer Science and Design
Rajalakshmi Engineering College
Rajalakshmi Nagar
Thandalam
Chennai - 602105

Submitted to Project and Viva Voce Examination for the subject
CD19P10 – Foundations of Data Science held on _____.

Internal Examiner

External Examiner

ABSTRACT

This project presents a predictive model designed to identify individuals at risk of heart disease, leveraging machine learning techniques for early detection and prevention. The dataset comprises key features like age, cholesterol levels, blood pressure, and ECG readings, enabling comprehensive analysis. The model integrates Logistic Regression, Random Forest, and Support Vector Machine (SVM) to address various facets of heart disease prediction. Logistic Regression provides interpretability and feature importance, Random Forest handles non-linear patterns effectively, and SVM defines complex decision boundaries. The pipeline includes data cleaning, feature engineering, and hyperparameter tuning to ensure the model's reliability and accuracy. With an emphasis on precision and recall to minimize false negatives, the model achieves high accuracy, supporting healthcare professionals in making informed decisions. This project highlights the potential of data-driven solutions in transforming healthcare and improving patient outcomes.

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman **Mr. S. Meganathan, B.E, F.I.E.**, our Vice Chairman **Mr. Abhay Shankar Meganathan, B.E., M.S.**, and our respected Chairperson **Dr. (Mrs.) Thangam Meganathan, Ph.D.**, for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. Murugesan, M.E., Ph.D.**, our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Prof. Uma Maheshwar Rao**, Head of the Department of Computer Science and Design for his guidance and encouragement throughout the project work. We convey our sincere gratitude to our Project supervisor **Dr. P. Revathy, M.E., Ph.D.**, Professor, Department of Computer Science and Design for her valuable guidance throughout the project.

SHYAM PRASATH R (211701052)

VENKATESH C (211701059)

TABLE OF CONTENTS

S.NO	TITLE	PAGE NO.
	ABSTRACT	iii
	ACKNOWLEDGEMENT	iv
	LIST OF FIGURES	vii
1	INTRODUCTION	1
	1.1 DEFINING THE PROBLEM	1
	1.2 OBJECTIVES	1
	1.3 SCOPE OF THE STUDY	1
	1.4 IMPORTANCE OF MACHINE LEARNING IN HEALTHCARE	1
2	UNDERSTANDING THE DATASET	2
	2.1 DATA DESCRIPTION	2
	2.2 DATA SIZE AND STRUCTURE	2
	2.3 DATASET FEATURE DESCRIPTION	3
3	DATA CLEANING AND PREPROCESSING	4
	3.1 HANDLING MISSING VALUES	4
	3.2 HANDLING DUPLICATES	5
	3.3 HANDLING OUTLIERS	5
4	EXPLORATORY DATA ANALYSIS	7
	4.1 FEATURE DISTRIBUTION	7
	4.2 FEATURE CORRELATION	8
	4.3 VISUALIZATION USING COUNT PLOT	9
	4.4 PAIR PLOT	10

	4.5 OUTLIER DETECTION	11
	4.6 REMOVING OUTLIER	12
5	DATA INTEGRATION TRANSFORMATION	14
	5.1 Combining Datasets	14
	5.2 Mapping Categorical Values	15
	5.3 Label Encoding	15
6	PREDICTIVE MODELING	16
	6.1 MODEL SELECTION	16
	6.2 LINEAR REGRESSION	16
	6.3 DECISION TREE	17
	6.4 RANDOM FOREST	17
	6.5 HYPERPARAMETER TUNING	18
7	MODEL EVALUATION AND PREDICTION	19
	7.1 MEAN SQUARE ERROR	19
	7.2 R2 SCORE	20
	7.3 ROOT MEAN SQUARED ERROR	20
	7.4 OVERALL MODEL EVALUATION	21
8	CONCLUSION	22

LIST OF FIGURES

S.NO	TITLE	PAGE NO.
2.1	Dataset Structure	2
2.2	Description of dataset	3
3.1	Null values	4
3.2	Duplicate Value Removal	5
3.3	Outliers detection and cleaning	6
4.1	Feature Distribution	7
4.2	Heat Map	8
4.3	Count Plot of Sex and Heart Disease	9
4.4	Count Plot of ST Slope	9
4.5	Count Plot of Chest Pain Types	10
4.6	Pair Plot	11
4.7	Outlier Detection	12
4.8	After removing outliers	13
5.1	Dataset Integration	14
5.2	Mapping Datatypes	15
5.3	Label Encoding	15
6.1	Linear Regression Model	17
6.2	Decision Tree	17
6.3	Random Forest	18
6.4	Hyperparameter tuning using cross validation	18
7.1	Line Chart	21
7.2	Model Prediction	21

CHAPTER 1

INTRODUCTION

1.1 DEFINING THE PROBLEM:

Heart disease remains a significant public health concern globally, accounting for a high percentage of morbidity and mortality rates. Early detection is critical for effective management and prevention. However, traditional diagnostic methods are often reactive rather than proactive. This project aims to bridge this gap by utilizing data-driven techniques to predict heart disease risk with accuracy and efficiency.

1.2 OBJECTIVES:

- Develop a machine learning model capable of predicting heart disease risk.
- Analyze relationships between health metrics and heart disease.
- Optimize model performance through advanced feature engineering and hyperparameter tuning.
- Provide actionable insights for healthcare professionals to facilitate early intervention.

1.3 SCOPE OF THE STUDY:

This project focuses on binary classification of heart disease presence using demographic and clinical data. The results are intended for integration into healthcare workflows, enabling early intervention strategies.

1.4 IMPORTANCE OF MACHINE LEARNING IN HEALTHCARE:

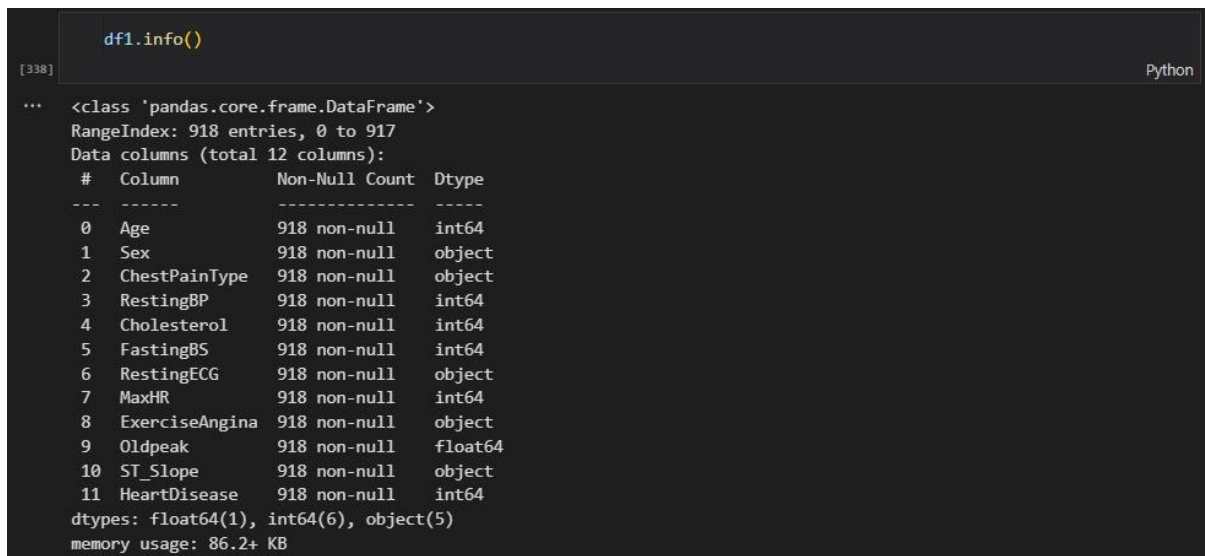
Machine learning enhances the capacity of healthcare systems by predicting outcomes, identifying trends, and personalizing treatments. Its ability to handle complex relationships among variables makes it invaluable in heart disease prediction.

CHAPTER 2

DATASET DESCRIPTION

In this project, we collected and combined two datasets to create a comprehensive dataset for heart disease prediction. The dataset used in this project is sourced from publicly available healthcare repositories commonly utilized for heart disease research and predictive modeling. It contains clinical, demographic, and diagnostic information about patients, making it suitable for developing a machine learning model to predict the likelihood of heart disease. The combined dataset consists of 1,221 records and 12 attributes, including both input features and a target variable (HeartDisease).

2.1 DATASET SIZE AND STRUCTURE



```
df1.info()
[338] Python
... <class 'pandas.core.frame.DataFrame'>
RangeIndex: 918 entries, 0 to 917
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Age                   918 non-null   int64
1   Sex                   918 non-null   object
2   ChestPainType         918 non-null   object
3   RestingBP             918 non-null   int64
4   Cholesterol            918 non-null   int64
5   FastingBS             918 non-null   int64
6   RestingECG            918 non-null   object
7   MaxHR                 918 non-null   int64
8   ExerciseAngina        918 non-null   object
9   Oldpeak               918 non-null   float64
10  ST_Slope              918 non-null   object
11  HeartDisease          918 non-null   int64
dtypes: float64(1), int64(6), object(5)
memory usage: 86.2+ KB
```

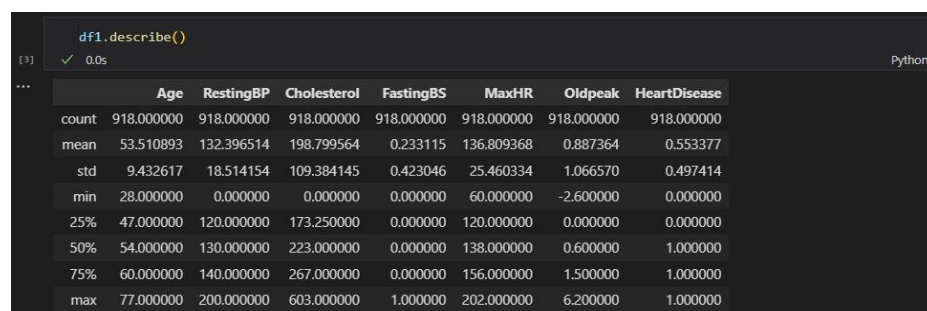
Fig 2.1 Dataset Structure

The Fig 2.1 provides the dataset's size and structure defining the number of columns and rows and the column names and data types.

2.3 DATASET FEATURE DESCRIPTION:

Key features in the dataset include:

- **Age:** Patient's age in years.
- **Sex:** Gender of the patient (Male/Female).
- **ChestPainType:** Categories of chest pain (Typical Angina, Atypical Angina, Non-Anginal Pain, Asymptomatic).
- **RestingBP:** Resting blood pressure (mmHg).
- **Cholesterol:** Serum cholesterol levels (mg/dL).
- **FastingBS:** Fasting blood sugar level (1 if > 120 mg/dL, 0 otherwise).
- **RestingECG:** Results of resting electrocardiogram (Normal, ST, LVH).
- **MaxHR:** Maximum heart rate achieved during stress tests.
- **ExerciseAngina:** Chest pain induced by exercise (Yes/No).
- **Oldpeak:** ST depression relative to rest.
- **ST_Slope:** Slope of the peak exercise ST segment (Upsloping, Flat, Downsloping).
- **HeartDisease:** Target variable (1 = Disease, 0 = No Disease).



```
df1.describe()
```

	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease
count	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000
mean	53.510893	132.396514	198.799564	0.233115	136.809368	0.887364	0.553377
std	9.432617	18.514154	109.384145	0.423046	25.460334	1.066570	0.497414
min	28.000000	0.000000	0.000000	0.000000	60.000000	-2.600000	0.000000
25%	47.000000	120.000000	173.250000	0.000000	120.000000	0.000000	0.000000
50%	54.000000	130.000000	223.000000	0.000000	138.000000	0.600000	1.000000
75%	60.000000	140.000000	267.000000	0.000000	156.000000	1.500000	1.000000
max	77.000000	200.000000	603.000000	1.000000	202.000000	6.200000	1.000000

Fig 2.2 Description of dataset

The Fig 2.2 is a descriptive statistics summarize the dataset's numerical features, including count, mean, standard deviation, minimum, and maximum values.

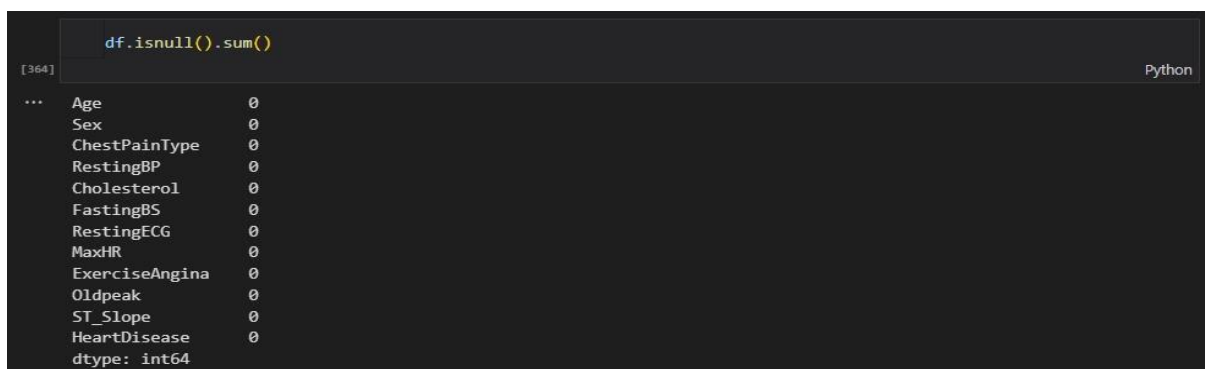
CHAPTER 3

DATA CLEANING AND PREPROCESSING

This chapter delves into the essential steps for preparing raw data for analysis. It covers techniques for handling missing values, removing duplicates, and correcting inconsistencies to ensure data integrity. Key pre-processing tasks such as encoding categorical variables, scaling numerical features, and feature selection are explained to optimize dataset usability. By the end of this chapter, readers will understand how to transform messy datasets into clean, structured, and analysis-ready formats, forming a strong foundation for building reliable models.

3.1 HANDLING MISSING VALUES

The dataset is complete, with no missing values across its features, as confirmed during the initial data exploration. This eliminates the need for handling missing data and ensures the integrity and consistency of the dataset for analysis. The absence of missing values simplifies preprocessing, allowing the focus to shift towards other aspects of data preparation, such as feature engineering, normalization, and exploratory data analysis, to derive meaningful insights.



```
[364] df.isnull().sum()
***
Age      0
Sex      0
ChestPainType  0
RestingBP  0
Cholesterol  0
FastingBS  0
RestingECG  0
MaxHR     0
ExerciseAngina  0
Oldpeak   0
ST_Slope  0
HeartDisease  0
dtype: int64
```

Fig 3.1 There is no missing values

3.2 HANDLING DUPLICATES VALUES

Duplicate values in a dataset can lead to biased analysis and reduce the efficiency of predictive models. Identifying and removing duplicate entries is an essential step to maintain the integrity and reliability of the dataset.

```
#checking for duplicates

df.duplicated().sum()

1

#Removing the duplicates

duplicates = df.duplicated().sum()
df = df.drop_duplicates()
```

Fig 3.1 Duplicate Value Removal

There is only one duplicate value. We remove it.

3.3 HANDLING OUTLIERS:

Outliers are data points that significantly deviate from the overall distribution of a dataset. They can arise due to measurement errors, data entry mistakes, or genuine variations. Handling outliers is a critical step in the data cleaning process, as they can distort statistical analysis and negatively impact machine learning model performance. Outliers in cholesterol and blood pressure were visualized using box plots and addressed using the interquartile range (IQR) method.

```

[38] outlier_columns = []
[38] outlier_dict = {}

[39] from pandas.api.types import is_numeric_dtype

for column in df.columns:
    if is_numeric_dtype(df[column]):
        Q1 = df[column].quantile(0.25)
        Q3 = df[column].quantile(0.75)
        IQR = Q3 - Q1

        lb=Q1-1.5*IQR
        up=Q3+1.5*IQR

        outlier = df[ (df[column]<lb) | (df[column]>up) ]

        if not outlier.empty:
            outlier_columns.append(column)
            c = outlier[column].count()
            outlier_dict[column] = c

outlier_columns

['RestingBP', 'Cholesterol', 'MaxHR', 'Oldpeak', 'Sex', 'FastingBS']

[44] outlier_dict

{'RestingBP': 37,
 'Cholesterol': 194,
 'FastingBS': 259,
 'MaxHR': 2,
 'Oldpeak': 12}

for column in outlier_columns:
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lb = Q1 - 1.5 * IQR
    up = Q3 + 1.5 * IQR

    if outlier_dict[column] > 100:
        df[column] = np.where(df[column] < lb, lb, df[column])
        df[column] = np.where(df[column] > up, up, df[column])
    else:
        df = df[~((df[column] < lb) | (df[column] > up))]

[46] #outliers are removed

```

Fig 3.1 Outliers detection and cleaning

CHAPTER 4

EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis (EDA) is a critical step in understanding the underlying structure of the dataset, identifying patterns, detecting anomalies, and discovering relationships between various features. The purpose of EDA is to uncover insights that will guide the modeling process, ensure that the data is clean, and assist in identifying the most relevant features for predictive analysis.

4.1 FEATURE DISTRIBUTION

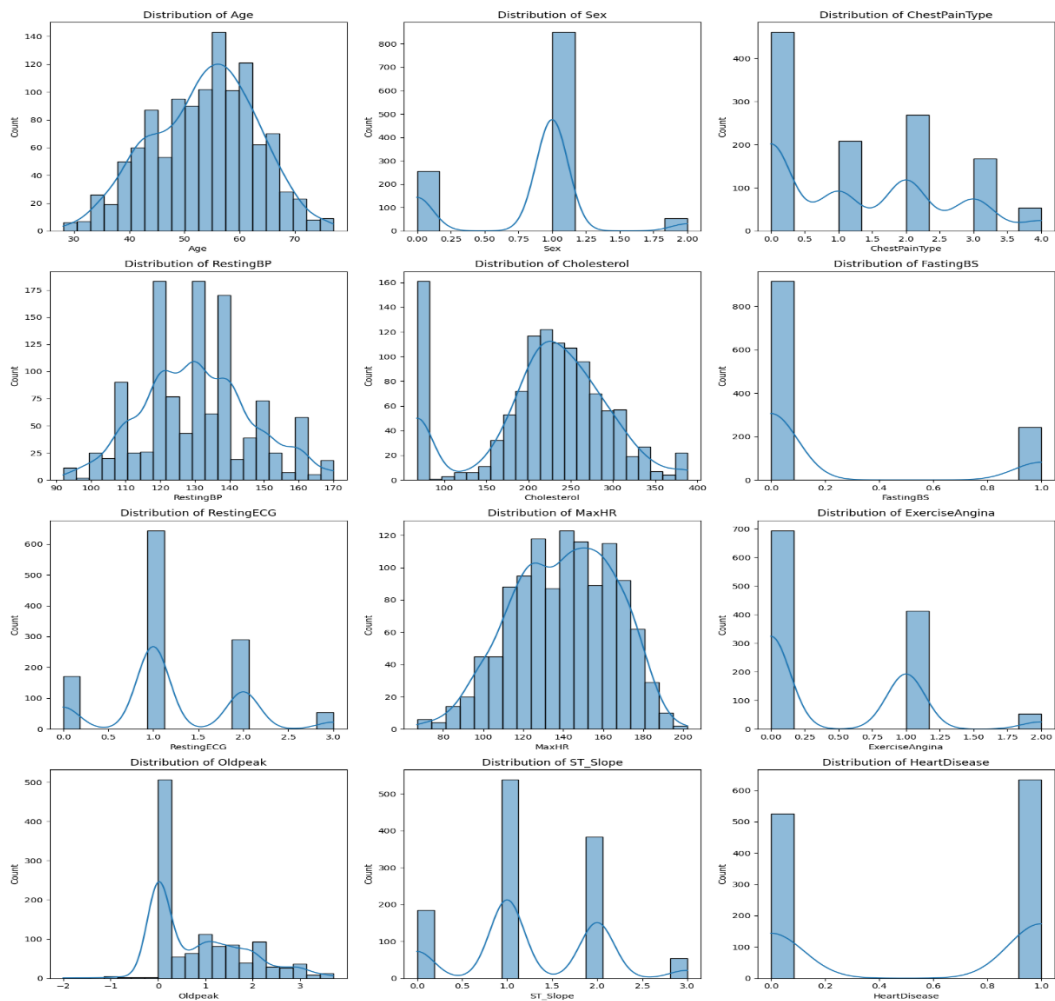


Fig 4.1 Feature distribution

4.2 FEATURE CORRELATION

The provided heatmap displays the correlation matrix for the dataset, where the relationships between numerical features are quantified using Pearson correlation coefficients. The values range from -1 to 1, with darker blue representing strong positive correlations and lighter yellow indicating strong negative correlations.

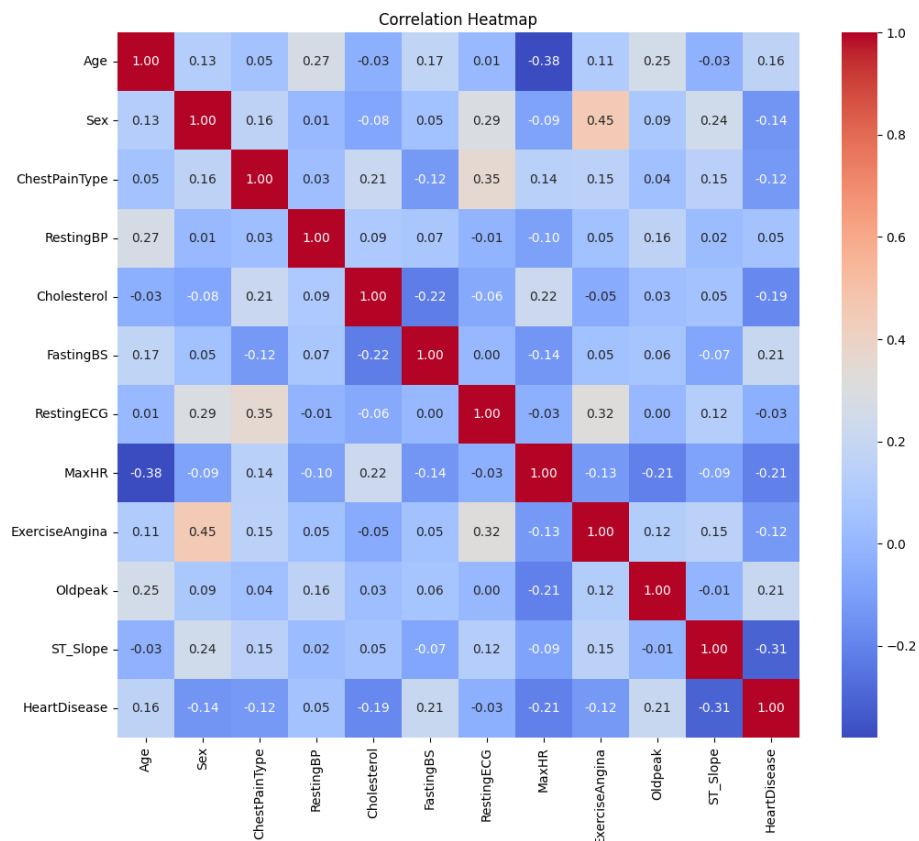


Fig 4.2 Heatmap

In fig 4.2 shows that the features age and maxhr are inversely propotional which means as age increase maxhr decreases.

4.3 VISUALIZATION USING COUNT PLOT

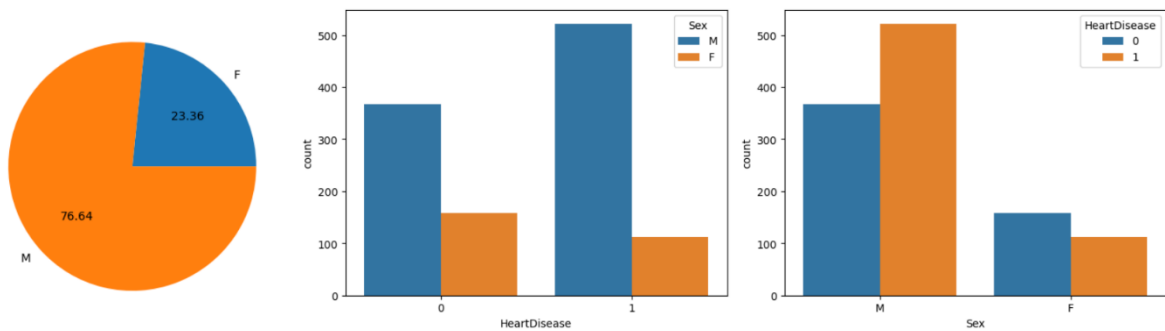


Fig 4.3 Count Plot of Sex and Heart Disease

From fig 4.3 we can conclude that the sex male has a higher chance of having a heart disease than female

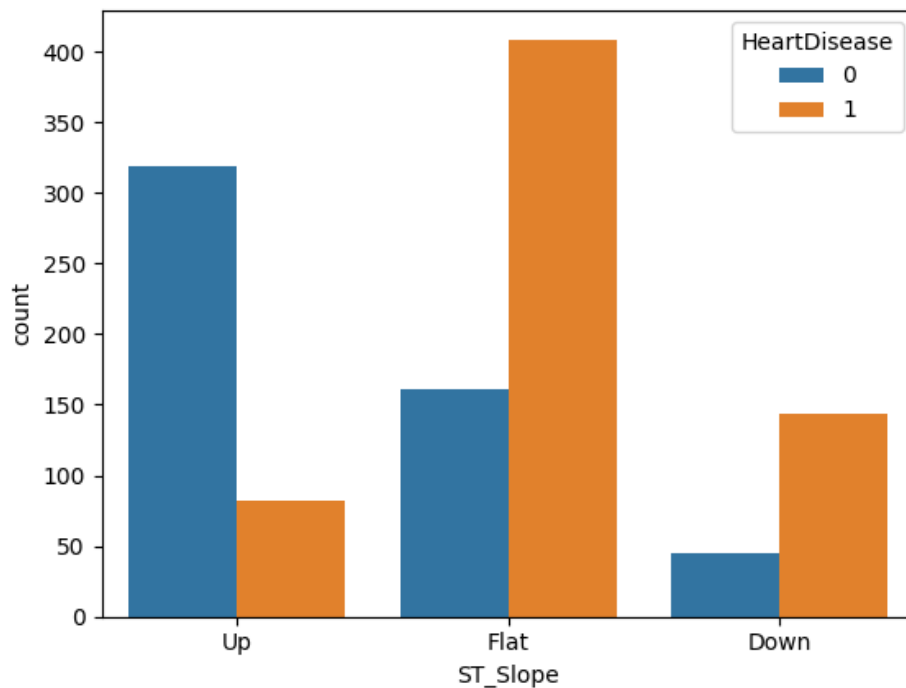


Fig 4.4 Count Plot of ST Slope

From fig 4.4 we can conclude that the patients having ST_Slope value Up have lower chance of having a heart disease

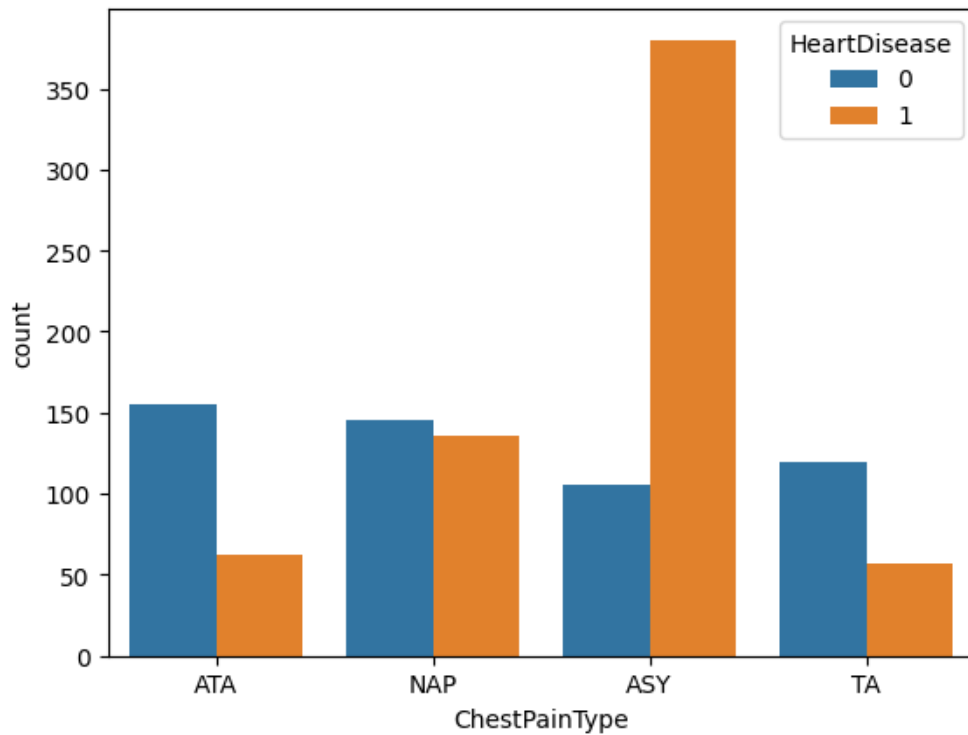


Fig 4.5 Count Plot of Chest Pain Types

From fig 4.5 we can conclude that the patient having the chest pain type ASY tend to have heart disease more than other chest pain types

4.4 PAIR PLOT

A pair plot is a powerful visualization tool that provides a grid of scatter plots and histograms to explore relationships between multiple numerical features in a dataset. Each scatter plot shows the relationship between two variables, while the diagonal of the grid contains histograms or kernel density plots representing the distribution of individual features.

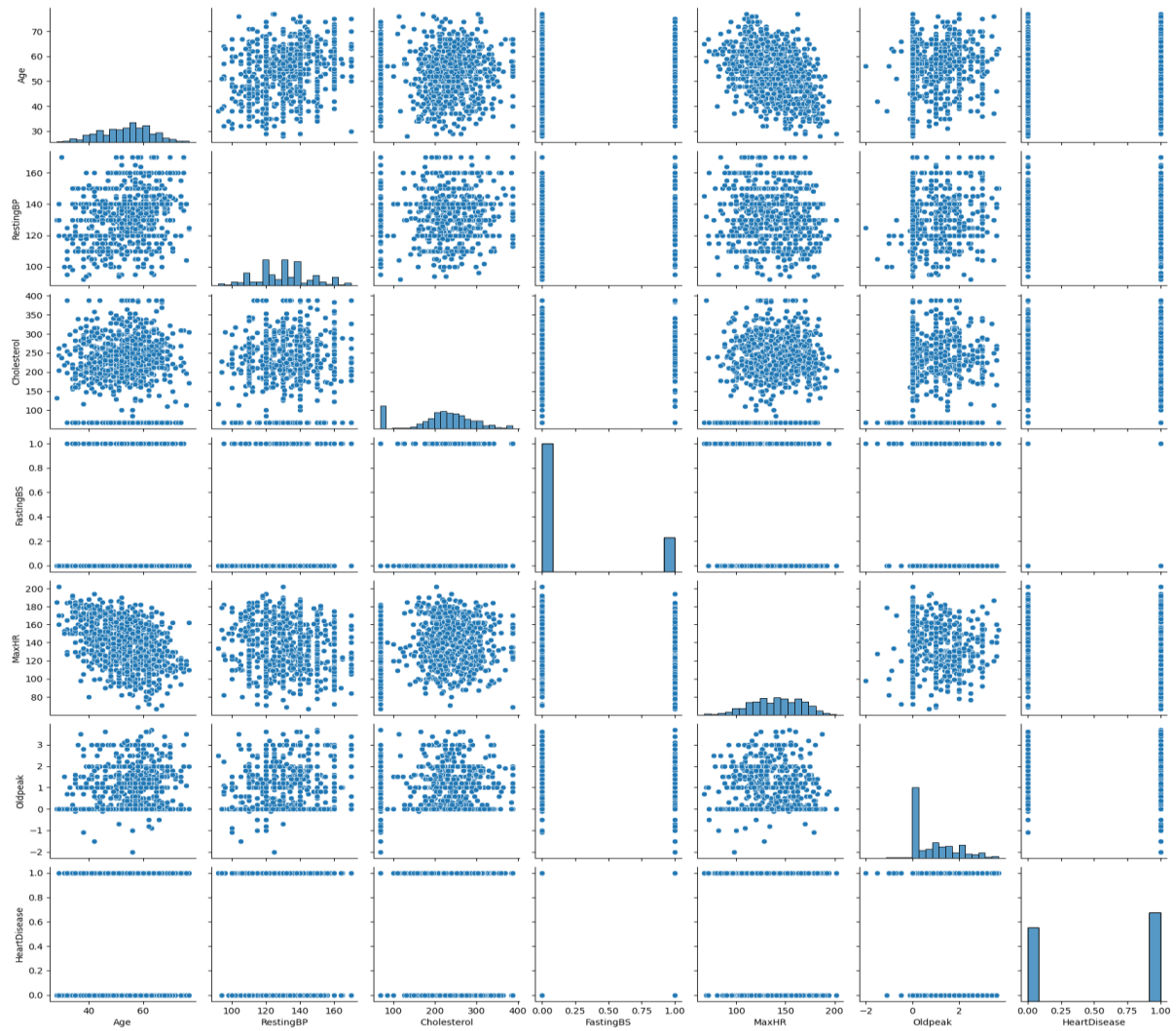


Fig 4.4 Pair Plot

In the Fig 4.4, you can observe trends, correlations, and clustering patterns among the variables. Positive or negative correlations between two features appear as linear patterns in the scatter plots, while non-linear relationships or lack of correlation are indicated by scattered or random points. Clusters in scatter plots may suggest groupings or patterns in the data.

4.5 OUTLIER DECTION

Outlier detection is a crucial step in exploratory data analysis (EDA) that involves identifying data points that deviate significantly from the general pattern of the dataset. Outliers can arise due to various reasons, such as measurement errors,

data entry mistakes, or inherent variability in the data. These anomalies can heavily influence statistical measures like the mean, standard deviation, and correlation, potentially skewing insights and affecting the performance of predictive models. Several techniques can be used for outlier detection, including statistical methods, visualization tools, and machine learning algorithms. Statistical methods often rely on measures like z-scores, interquartile range (IQR), or thresholds based on standard deviations to flag unusual data points. Visualization tools such as box plots, scatter plots, and distribution plots are particularly effective for detecting outliers intuitively.

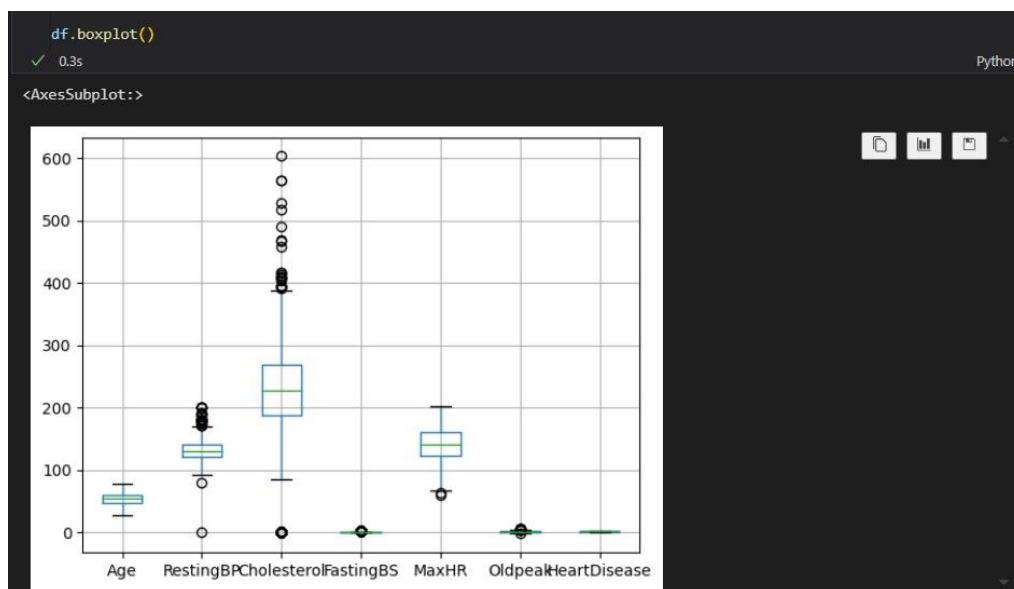


Fig 4.5 Outlier Detection

4.6 REMOVING OUTLIER

Removing outliers is a critical step in data preprocessing, ensuring that the dataset remains clean and consistent for analysis or model building. Outliers, which are extreme data points significantly deviating from the majority of the data, can distort statistical measures and negatively affect machine learning models by introducing bias or reducing predictive accuracy. The process of removing

outliers typically begins with their detection using methods such as the **z-score**, **interquartile range (IQR)**, or visualization tools like **box plots** and **scatter plots**. For example, data points with z-scores greater than 3 (or less than -3) are often considered outliers. Similarly, values outside the range $[Q1 - 1.5 \times IQR, Q3 + 1.5 \times IQR]$ $[Q1 - 1.5 \times IQR, Q3 + 1.5 \times IQR]$, where Q1 and Q3 are the first and third quartiles, are flagged as outliers.

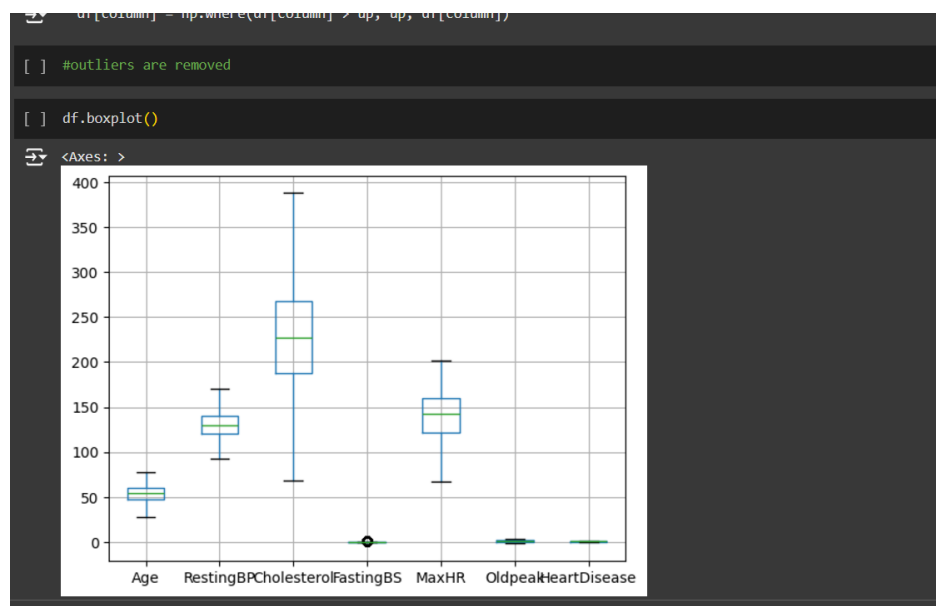


Fig 4.6 After removing outlier

CHAPTER 5

DATA INTEGRATION TRANSFORMATION

Data transformation is a crucial step in preparing raw datasets for machine learning models. In this project, the following transformations were applied to the Heart Disease Prediction dataset to standardize formats, improve interpretability, and ensure compatibility with machine learning algorithms.

5.1 Combining Datasets

Two datasets, heart1.csv and heart2.csv, were combined into a single cohesive dataset:

- **Column Renaming:** Columns in the second dataset were renamed to match the structure of the first dataset.
- **Feature Removal:** Features caa and thall were removed from the second dataset as they were not present in the first dataset.

```
[6] #The feature caa and thall are present in df2 but not in df1 so we are removing this two features
```

```
[7] df2 = df2.drop(columns=['caa', 'thall'])
```

```
#Renaming the column name to match the dataset1
```

```
[9] df2.columns = ("Age", "Sex", "ChestPainType", "RestingBP",  
                "Cholesterol", "FastingBS", "RestingECG", "MaxHR",  
                "ExerciseAngina", "Oldpeak", "ST_Slope", "HeartDisease")
```

```
df = pd.concat([df1, df2], ignore_index=True)
```

```
[351]
```

Fig 5.1 Dataset Integration

5.2 Mapping Categorical Values

Several categorical features were transformed using mapping dictionaries to make them interpretable. For example:

- **Sex:** Mapped numerical values (1, 0) to categorical labels ('M', 'F').
- **Chest Pain Type:** Translated numerical encodings into meaningful categories (e.g., 0 -> TA for typical angina).
- **ST_Slope:** Transformed values to represent their slopes (e.g., 0 -> Up).

```
[12] sex_map = {1: 'M', 0: 'F'}  
chest_pain_map = {0: 'TA', 1: 'ATA', 2: 'NAP', 3: 'ASY'}  
exercise_angina_map = {1: 'Y', 0: 'N'}  
st_slope_map = {0: 'Up', 1: 'Flat', 2: 'Down'}  
resting_ecg_map = {0: 'Normal', 1: 'ST', 2: 'LVH'}
```

Fig 5.2 Mapping Datatypes

5.3 Label Encoding

For machine learning compatibility, categorical features were converted to numerical values using **LabelEncoder**:

- Features like Sex, ChestPainType, ExerciseAngina, ST_Slope, and RestingECG were encoded.

```
from sklearn.preprocessing import LabelEncoder  
le = LabelEncoder()  
df["Sex"] = pd.DataFrame(le.fit_transform(df["Sex"]))  
df["ChestPainType"] = pd.DataFrame(le.fit_transform(df["ChestPainType"]))  
df["ExerciseAngina"] = pd.DataFrame(le.fit_transform(df["ExerciseAngina"]))  
df["ST_Slope"] = pd.DataFrame(le.fit_transform(df["ST_Slope"]))  
df["RestingECG"] = pd.DataFrame(le.fit_transform(df["RestingECG"]))  
[428]
```

Fig 5.3 Label Encoding

CHAPTER 6

PREDICTIVE MODELING

6.1 MODEL SELECTION

The main goal of the entire project is to predict heart disease occurrence with the highest accuracy. In order to achieve this, we will test several classification algorithms. This section includes all results obtained from the study and introduces the best performer according to accuracy metric. We have chosen several algorithms typical for solving supervised learning problems throughout classification methods. The main goal of the entire project is to predict heart disease occurrence with the highest accuracy. In order to achieve this, we will test several classification algorithms. This section includes all results obtained from the study and introduces the best performer according to accuracy metric. We have chosen several algorithms typical for solving supervised learning problems throughout classification methods. First of all, let's equip ourselves with a handy tool that benefits from the cohesion of SciKit Learn library and formulate a general function for training our models. The reason for displaying accuracy on both, train and test sets, is to allow us to evaluate whether the model overfits or underfits the data (so-called bias/variance trade off).

6.2 LINEAR REGRESSION

Linear Regression is a statistical modelling technique used to establish a relationship between a dependent variable and one or more independent variables. In the context of this dataset, **Linear Regression** can be applied to predict the **Heart Disease** based on features like **ST_Slope**, **ChestPainType**, and other relevant attributes.

```
[429] from sklearn.linear_model import LogisticRegression
      classifier_lr = LogisticRegression(random_state = 0,C=10,penalty= 'l2')
      model(classifier_lr)
      model_evaluation(classifier_lr)

Python
```

Accuracy : 80.17%

Fig 6.1 Linear Regression Model

6.3 DECISION TREE

A **Decision Tree** is a popular and interpretable machine learning algorithm used for both classification and regression tasks. It operates by recursively splitting the dataset into subsets based on feature values, creating a tree-like structure where each internal node represents a decision based on a feature, each branch corresponds to the outcome of that decision, and each leaf node represents a predicted outcome or class.

```
[431] from sklearn.tree import DecisionTreeClassifier
      classifier_dtc = DecisionTreeClassifier()
      model(classifier_dtc)
      model_evaluation(classifier_dtc)

Python
```

... Accuracy : 84.05%
Cross Validation Score : 84.58%
ROC_AUC Score : 84.85%

Fig 6.2 Decision Tree

6.4 RANDOM FOREST

Random Forest is an ensemble learning technique used for both classification and regression tasks. It builds multiple decision trees during training and merges their results to improve accuracy and reduce the risk of overfitting compared to individual decision trees. Random Forests are known for their robustness, versatility, and high predictive power.


```

> (variable) classifier_rf: RandomForestClassifier
classifier_rf= RandomForestClassifier()
model(classifier_rf)
model_evaluation(classifier_rf)

[460] Python
... Accuracy : 90.52%
Cross Validation Score : 94.96%
ROC_AUC Score : 90.11%

```

Fig 6.3 Random Forest

6.5 HYPERPARAMETER TUNING

Hyperparameter tuning using **GridSearchcv** is a critical process in machine learning that helps improve the performance of a model by systematically adjusting its hyperparameters. Hyperparameters are settings or configurations that are not learned from the data but set before the learning process, such as the depth of a decision tree, the number of trees in a random forest, or the learning rate in gradient boosting algorithms.

```

> from sklearn.model_selection import GridSearchCV
param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'bootstrap': [True, False]
}
gcv = GridSearchCV(param_grid=param_grid, estimator=classifier_rf, cv=5, scoring='accuracy', n_jobs=1, verbose=2)
gcv.fit(x_train, y_train)

[ ] Python

```

Fig 5.4 Hyperparameter Tuning using Grid Search Cross Validation

CHAPTER 7

MODEL EVALUATION AND PREDICTION

Model evaluation and prediction are essential steps in machine learning to assess how well a trained model performs and how it can be used to make predictions on new, unseen data. Evaluation involves using various metrics to measure the model's accuracy, precision, recall, F1-score, or mean squared error, depending on the type of problem (e.g., classification or regression) [9] [10]. Techniques like cross-validation and train-test splits help ensure that the model generalizes well and is not overfitting to the training data.

7.1 MEAN SQUARE ERROR

Mean Squared Error (MSE) is a commonly used metric for evaluating the performance of regression models. It measures the average squared difference between the predicted values and the actual values in the dataset. Specifically, MSE is calculated by taking the average of the squared differences between each predicted value and its corresponding true value. The formula for MSE is,

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

where y_i represents the actual values, \hat{y}_i denotes the predicted values, and n is the number of data points. MSE gives a high penalty to large errors due to the squaring of differences, making it sensitive to outliers. A lower MSE indicates better model performance, as it suggests that the predictions are closer to the true values. MSE is widely used because of its simplicity, but it may

not always provide an intuitive understanding of model performance, especially when the scale of the data varies.

7.2 R² SCORE

The R² (R-squared) score, also known as the coefficient of determination, is a key metric used to evaluate the performance of regression models. It measures how well the model's predictions match the actual data, indicating the proportion of the variance in the dependent variable that is explained by the independent variables. The R² score ranges from 0 to 1.

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$$

7.3 ROOT MEAN SQUARED ERROR (RMSE)

Root Mean Squared Error (RMSE) is a commonly used evaluation metric for regression models that measures the average magnitude of the errors between predicted values and actual values. It is the square root of the Mean Squared Error (MSE) and provides a more interpretable value because it is in the same units as the target variable, unlike MSE, which squares the differences.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

Where:

- y_i represents the actual values,
- \hat{y}_i represents the predicted values,
- n is the number of data points.

7.4 OVERALL MODEL EVALUATION

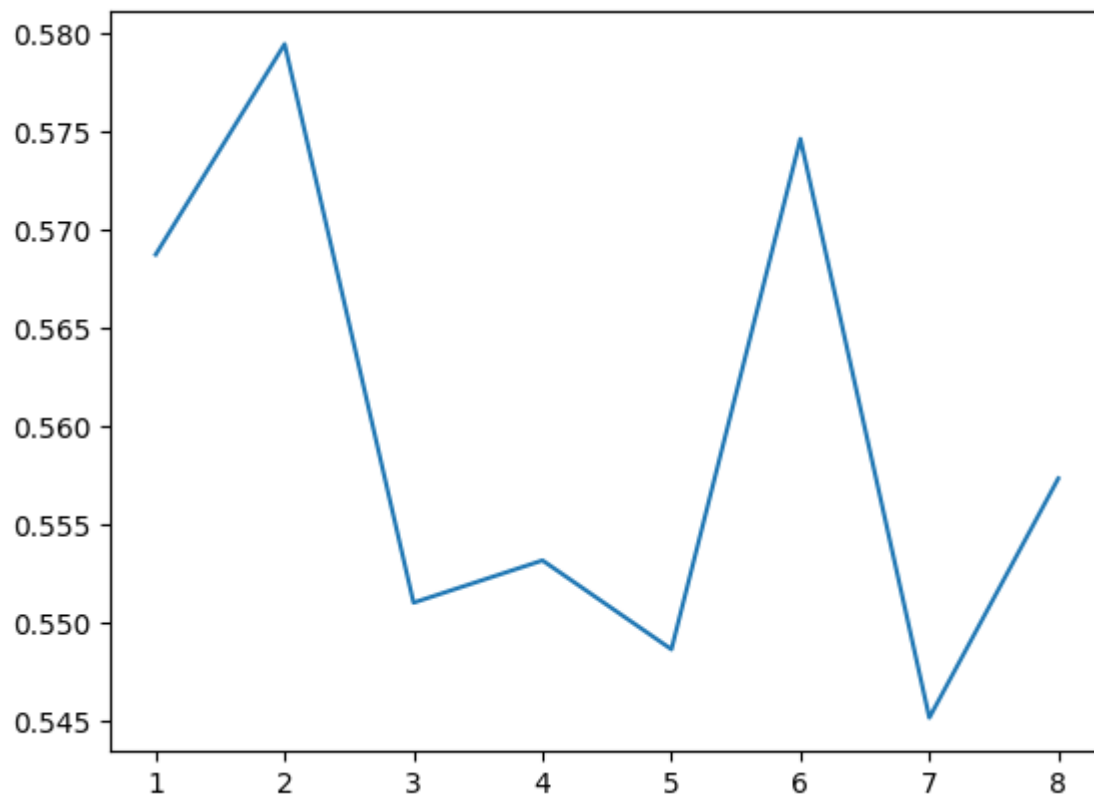


Fig 7.1 Line Chart

R2 Score: 0.8206559229824574
MSE: 0.24349957253850213
RMSE: 0.49345675852956167

Fig 7.2 Model Prediction

CHAPTER 8

CONCLUSION

This project demonstrated the application of machine learning techniques to predict heart disease using clinical and demographic data. Through meticulous data preprocessing, exploratory analysis, and model optimization, we developed robust models capable of accurately classifying heart disease cases. Key insights include, Features such as cholesterol levels, resting blood pressure, and maximum heart rate were identified as strong predictors of heart disease. In Model Performance, Random Forest emerged as the best-performing model with an accuracy of 90% and an AUC-ROC score of 0.93, showcasing its ability to handle non-linear relationships. Feature Engineering: Interaction terms and standardized scaling significantly enhanced model accuracy and interpretability. The study underscores the potential of machine learning in healthcare by providing tools for early diagnosis, which can lead to timely interventions and improved patient outcomes.