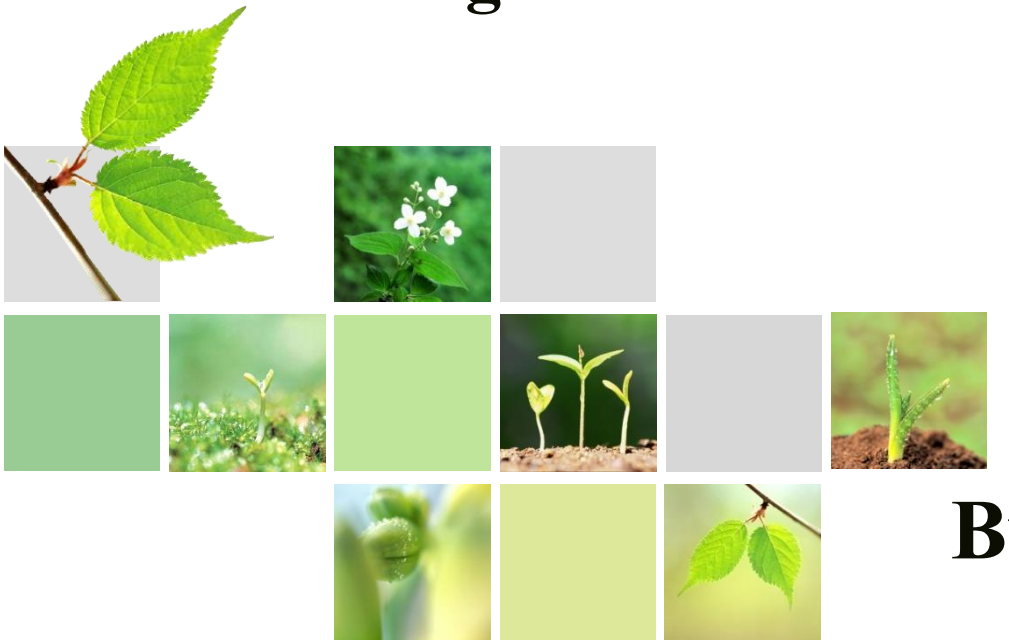
Basic Concept of Classification (Data Mining)

Predicting Heart Disease Using Machine Learning



By : Ali pourfereydoon



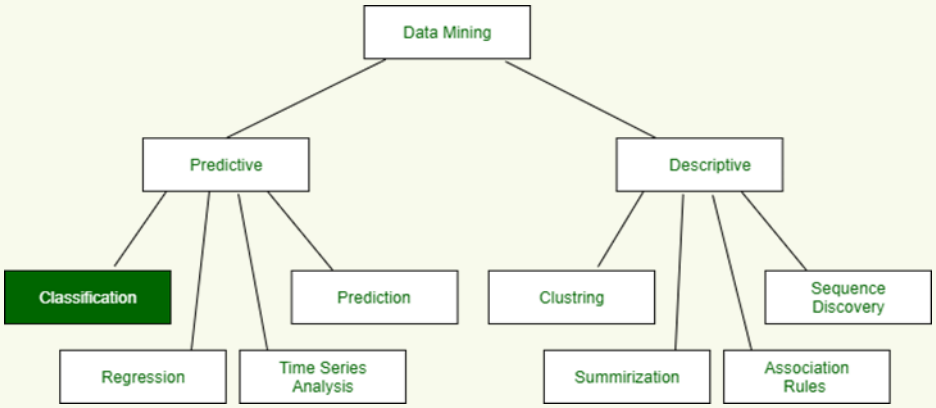
Chapter 1

What is a Data Mining and classification:

Chapter 1

introduction

Data mining in general terms means mining or digging deep into data that is in different forms to gain patterns, and to gain knowledge on that pattern. In the process of data mining, large data sets are first sorted, then patterns are identified and relationships are established to perform data analysis and solve problems.



Chapter 1

introduction

Classification:

Classification is a widely used technique in data mining and is applied in a variety of domains, such as email filtering, sentiment analysis, and medical diagnosis.

Classification is a task in data mining that involves assigning a class label to each instance in a dataset based on its features. The goal of classification is to build a model that accurately predicts the class labels of new instances based on their features.

Chapter 1

introduction

There are two main types of classification:

binary classification and multi-class classification. Binary classification involves classifying instances into two classes, such as "spam" or "not spam", while multi-class classification involves classifying instances into more than two classes.

introduction

Binary: Possesses only two values i.e. True or False

Example: Suppose there is a survey evaluating some products. We need to check whether it's useful or not

Nominal: When more than two outcomes are possible. It is in Alphabet form rather than being in Integer form.

Example: One needs to choose some material but of different colors. So, the color might be Yellow, Green, Black, Red.

Different Colors: Red, Green, Black, Yellow

introduction

Ordinal: Values that must have some meaningful order.

Example: Suppose there are grade sheets of few students which might contain different grades as per their performance such as A, B, C, D

Continuous: May have an infinite number of values, it is in float type

Example: Measuring the weight of few Students in a sequence or orderly manner i.e. 50, 51, 52, 53

Weight: 50, 51, 52, 53

Discrete: Finite number of values.

Example: Marks of a Student in a few subjects: 65, 70, 75, 80, 90

Marks: 65, 70, 75, 80, 90



Chapter 2

The process of building a classification model typically involves the following steps:

Chapter 2

Data Collection:

The first step in building a classification model is data collection. In this step, the data relevant to the problem at hand is collected. The data should be representative of the problem and should contain all the necessary attributes and labels needed for classification. The data can be collected from various sources, such as surveys, questionnaires, websites, and databases.

Data Preprocessing:

The second step in building a classification model is data preprocessing. The collected data needs to be preprocessed to ensure its quality. This involves handling missing values, dealing with outliers, and transforming the data into a format suitable for analysis. Data preprocessing also involves converting the data into numerical form, as most classification algorithms require numerical input.

Data Preprocessing:

Handling Missing Values: Missing values in the dataset can be handled by replacing them with the mean, median, or mode of the corresponding feature or by removing the entire record.

Dealing with Outliers: Outliers in the dataset can be detected using various statistical techniques such as z-score analysis, boxplots, and scatterplots. Outliers can be removed from the dataset or replaced with the mean, median, or mode of the corresponding feature.

Data Preprocessing:

Data Transformation: Data transformation involves scaling or normalizing the data to bring it into a common scale. This is done to ensure that all features have the same level of importance in the analysis.

Feature Selection:

The third step in building a classification model is feature selection. Feature selection involves identifying the most relevant attributes in the dataset for classification. This can be done using various techniques, such as correlation analysis, information gain, and principal component analysis.

Feature Selection:

Correlation Analysis: Correlation analysis involves identifying the correlation between the features in the dataset. Features that are highly correlated with each other can be removed as they do not provide additional information for classification.

Information Gain: Information gain is a measure of the amount of information that a feature provides for classification. Features with high information gain are selected for classification.

Chapter 2

Model Selection:

The fourth step in building a classification model is model selection. Model selection involves selecting the appropriate classification algorithm for the problem at hand. There are several algorithms available, such as :

Classifiers Of Machine Learning:

- Decision Trees
- Bayesian Classifiers
- Neural Networks
- K-Nearest Neighbour
- Support Vector Machines
- Linear Regression
- Logistic Regressio

Model Selection:

Classifiers can be categorized into two major types:

1.Discriminative

2. Generative:

Model Selection:

Discriminative: It is a very basic classifier and determines just one class for each row of data. It tries to model just by depending on the observed data, depends heavily on the quality of data rather than on distributions.
Example: Logistic Regression

Generative: It models the distribution of individual classes and tries to learn the model that generates the data behind the scenes by estimating assumptions and distributions of the model. Used to predict the unseen data.

Model Selection:

Example: Naive Bayes Classifier

Detecting Spam emails by looking at the previous data. Suppose 100 emails and that too divided in 1:4 i.e. Class A: 25%(Spam emails) and Class B: 75%(Non-Spam emails).

Now if a user wants to check that if an email contains the word cheap, then that may be termed as Spam. It seems to be that in Class A(i.e. in 25% of data), 20 out of 25 emails are spam and rest not.

And in Class B(i.e. in 75% of data), 70 out of 75 emails are not spam and rest are spam.

So, if the email contains the word cheap, what is the probability of it being spam ?? (= 80%).

Model Selection:

Decision Trees: Decision trees are a simple yet powerful classification algorithm. They divide the dataset into smaller subsets based on the values of the features and construct a tree-like model that can be used for classification.

Model Selection:

Support Vector Machines: Support Vector Machines (SVMs) are a popular classification algorithm used for both linear and nonlinear classification problems. SVMs are based on the concept of maximum margin, which involves finding the hyperplane that maximizes the distance between the two classes.

Chapter 2

Model Selection:

Neural Networks:

Neural Networks are a powerful classification algorithm that can learn complex patterns in the data. They are inspired by the structure of the human brain and consist of multiple layers of interconnected nodes

Model Training:

The fifth step in building a classification model is model training.

Model training involves using the selected classification algorithm to learn the patterns in the data. The data is divided into a training set and a validation set. The model is trained using the training set, and its performance is evaluated on the validation set.

Model Evaluation:

The sixth step in building a classification model is model evaluation.

Model evaluation involves assessing the performance of the trained model on a test set. This is done to ensure that the model generalizes well

Chapter 3

Real-Life Examples :

Chapter 3

Market Basket Analysis:

It is a modeling technique that has been associated with frequent transactions of buying some combination of items.

Example: Amazon and many other Retailers use this technique. While viewing some products, certain suggestions for the commodities are shown that some people have bought in the past.

Chapter 3

Weather Forecasting:

Changing Patterns in weather conditions needs to be observed based on parameters such as temperature, humidity, wind direction. This keen observation also requires the use of previous records in order to predict it accurately.

Chapter 3

Predicting Heart Disease Using Machine Learning

Goal:To build a classification model that predicts whether a patient has heart disease based on medical attributes.

Why classification : The target variable (target) has two classes:

0 → No heart disease

1 → Heart disease

This makes it a binary classification problem.

Chapter 3

Predicting Heart Disease Using Machine Learning
Step :1

STEP 1: Kaggle Dataset Import Setup

```
import kagglehub
redwankarimsony_heart_disease_data_path = kagglehub.dataset_download(
    'redwankarimsony/heart-disease-data'
)

print('Data source import complete.')
```

Chapter 3

Predicting Heart Disease Using Machine Learning

Step :1

STEP 1: Kaggle Dataset Import Setup

Explanation:

kagglehub is a library that allows downloading datasets directly from Kaggle.

The dataset heart-disease-data is downloaded from Kaggle.This ensures the dataset source is Kaggle.

The printed message confirms successful download.

Purpose: To legally and reproducibly import the dataset from Kaggle.

Predicting Heart Disease Using Machine Learning

Step :2

STEP 2: Import Required Libraries

```
import numpy as np
import pandas as pd
```

Explanation:

NumPy (np): used for numerical computations.

Pandas (pd): used for reading CSV files and handling tabular data.

Purpose: These libraries are fundamental for data preprocessing and analysis

Predicting Heart Disease Using Machine Learning
Step :2

STEP 2: Import Required Libraries

NumPy (np):

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

Predicting Heart Disease Using Machine Learning

Step :2

STEP 2: Import Required Libraries

NumPy (np):

There are several important differences between NumPy arrays and the standard Python sequences:

NumPy arrays have a fixed size at creation, unlike Python lists (which can grow dynamically).

The elements in a NumPy array are all required to be of the same data type, and thus will be the same size in memory.

Predicting Heart Disease Using Machine Learning
Step :2

STEP 2: Import Required Libraries

NumPy (np):

NumPy arrays facilitate advanced mathematical and other types of operations on large numbers of data. Typically, such operations are executed more efficiently and with less code than is possible using Python's built-in sequences.

A growing plethora of scientific and mathematical Python-based packages are using NumPy arrays; though these typically support Python-sequence input, they convert such input to NumPy arrays prior to processing, and they often output NumPy arrays.

Predicting Heart Disease Using Machine Learning

Step :2

STEP 2: Import Required Libraries

Pandas Introduction :

Pandas is an open-source Python library used for data manipulation, analysis and cleaning. It provides fast and flexible tools to work with tabular data, similar to spreadsheets or SQL tables.

Pandas is used in data science, machine learning, finance, analytics and automation because it integrates smoothly with other libraries such as:

NumPy: numerical operations

Matplotlib and Seaborn: data visualization

SciPy: statistical analysis

Scikit-learn: machine learning workflows

Predicting Heart Disease Using Machine Learning

Step :2

STEP 2: Import Required Libraries

Pandas Introduction :

Installation: Before using Pandas, make sure it is installed

```
pip install pandas
```

After the Pandas have been installed in the system we need to import the library. This module is imported using:

```
import pandas as pd
```

Note: pd is just an alias for Pandas. It's not required but using it makes the code shorter when calling methods or properties.

Predicting Heart Disease Using Machine Learning

Step :2

STEP 2: Import Required Libraries

Data Structures in Pandas:

Pandas provides two data structures for manipulating data which are as follows:

1. Pandas Series :

A Pandas Series is one-dimensional labeled array capable of holding data of any type (integer, string, float, Python objects etc.). The axis labels are collectively called indexes. Series is created by loading the datasets from existing storage which can be a SQL database, a CSV file or an Excel file.

Chapter 3

Predicting Heart Disease Using Machine Learning

Step :2

STEP 2: Import Required Libraries

Data Structures in Pandas:

```
import pandas as pd
import numpy as np

s = pd.Series()
print("Pandas Series: ", s)
data = np.array(['g', 'e', 'e', 'k', 's'])

s = pd.Series(data)
print("Pandas Series:\n", s)
```

Output

```
Pandas Series:  Series([], dtype: object)
Pandas Series:
 0    g
1    e
2    e
3    k
4    s
dtype: object
```

Predicting Heart Disease Using Machine Learning

Step :2

STEP 2: Import Required Libraries

Data Structures in Pandas:

2. Pandas DataFrame:

Pandas DataFrame is a two-dimensional data structure with labeled axes (rows and columns). It is created by loading the datasets from existing storage which can be a SQL database, a CSV file or an Excel file. It can be created from lists, dictionaries, a list of dictionaries etc.

Predicting Heart Disease Using Machine Learning

Step :2

STEP 2: Import Required Libraries

Data Structures in Pandas:

2. Pandas DataFrame:

```
import pandas as pd

df = pd.DataFrame()
print(df)
lst = ['Geeks', 'For', 'Geeks', 'is', 'portal', 'for', 'Geeks']

df = pd.DataFrame(lst)
print(df)
```

Output:

```
Empty DataFrame
Columns: []
Index: []
        0
0   Geeks
1     For
2   Geeks
3      is
4  portal
5     for
6   Geeks
```

Pandas DataFrame

Predicting Heart Disease Using Machine Learning

Step :2

STEP 2: Import Required Libraries

Data Structures in Pandas:

Operations in Pandas:

Pandas provides essential operations for working with structured data efficiently. The sections below introduce the most commonly used functionalities with short explanations and simple examples.

1. Loading Data: This operation reads data from files such as CSV, Excel or JSON into a DataFrame.

Predicting Heart Disease Using Machine Learning

Step :2

STEP 2: Import Required Libraries

Data Structures in Pandas:

Operations in Pandas:

```
import pandas as pd

df = pd.read_csv("data.csv")
print(df.head())
```

Output

	name	age	city	category	sales	a	b
0	John	25	Delhi	A	200	5	10
1	Riya	22	Mumbai	B	120	2	3
2	Aman	30	Pune	A	250	7	8
3	Neha	27	Kolkata	C	300	4	6
4	Arjun	35	Jaipur	B	200	1	9

Explanation: pd.read_csv("data.csv") reads the CSV file and loads it into a DataFrame and df.head() shows the first 5 rows of the data.

Predicting Heart Disease Using Machine Learning

Step :2

STEP 2: Import Required Libraries

Data Structures in Pandas:

Operations in Pandas:

2. Viewing and Exploring Data: After loading data, it is important to understand its structure and content. This methods allow you to inspect rows, summary statistics and metadata.

Chapter 3

Predicting Heart Disease Using Machine Learning

Step :2

STEP 2: Import Required Libraries

2. Viewing and Exploring Data:

```
print(df.info())
```

Output

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   name      5 non-null      object
 1   age       5 non-null      int64
 2   city      5 non-null      object
 3   category  5 non-null      object
 4   sales     5 non-null      int64
 5   a         5 non-null      int64
 6   b         5 non-null      int64
dtypes: int64(4), object(3)
memory usage: 412.0+ bytes
None
```

Predicting Heart Disease Using Machine Learning Step :2

STEP 2: Import Required Libraries

3. Handling Missing Data: Datasets often contain empty or missing values. Pandas provides functions to detect, remove or replace these values.

Explanation:

df.fillna(0)replaces missing values with 0

```
print(df.isnull().sum())
df = df.fillna(0)
```

Output

```
name        0
age         0
city        0
category    0
sales       0
a           0
b           0
dtype: int64
```

No Columns have NAN value

Predicting Heart Disease Using Machine Learning

Step :2

STEP 2: Import Required Libraries

4. Selecting and Filtering Data: This operation retrieves specific columns, rows or records that match a condition. It allows precise extraction of required information.

```
ages = df[df['age'] > 25]
print(ages)
```

Output

	name	age	city	category	sales	a	b
2	Aman	30	Pune	A	250	7	8
3	Neha	27	Kolkata	C	300	4	6
4	Arjun	35	Jaipur	B	200	1	9

Output of Filtering Data

Explanation df[df['age'] > 25] returns rows where the "age" value is greater than 25.

Predicting Heart Disease Using Machine Learning

Step :2

STEP 2: Import Required Libraries

5. Adding and Removing Columns: You can create new columns based on existing ones or delete unwanted columns from the DataFrame

```
df['total'] = df['a'] + df['b']
print(df.head())
```

Output

	name	age	city	category	sales	a	b	total
0	John	25	Delhi	A	200	5	10	15
1	Riya	22	Mumbai	B	120	2	3	5
2	Aman	30	Pune	A	250	7	8	15
3	Neha	27	Kolkata	C	300	4	6	10
4	Arjun	35	Jaipur	B	200	1	9	10

Adding new column "total"

Explanation: df['total'] = df['a'] + df['b'] creates a new column named "total".

Predicting Heart Disease Using Machine Learning Step :2

STEP 2: Import Required Libraries

6. Grouping Data (GroupBy): Grouping allows you to organize data into categories and compute values for each group for example, sums, counts or averages.

```
res = df.groupby('category')['sales'].sum()
print(res)
```

Output

```
category
A    450
B    320
C    300
Name: sales, dtype: int64
```

Grouping Data

Explanation: df.groupby('category') divides the dataset based on the "category" column.

Predicting Heart Disease Using Machine Learning
Step :3

STEP 3: Check Kaggle Input Directory

```
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

Explanation:

This code lists all files available in Kaggle's /kaggle/input directory.It helps confirm the dataset file path and name.

Purpose: To verify that the dataset exists and identify its exact location.

Predicting Heart Disease Using Machine Learning
Step :3

STEP 3: Check Kaggle Input Directory

What is os :

The Python os module provides tools for using operating system-dependent functionality, like reading or writing to the file system. It allows you to interface with the underlying operating system in a portable way.

Key Features:

Interacts with the operating system

Manipulates file paths and directories

Provides access to environment variables

Facilitates process management

Predicting Heart Disease Using Machine Learning

Step :3

STEP 3: Check Kaggle Input Directory

Object	Type	Description
os.path	Module	Provides common pathname manipulations
os.environ	Mapping	Gives access to environment variables
os.listdir()	Function	Lists directory contents
os.mkdir()	Function	Creates a directory
os.remove()	Function	Removes a file
os.rename()	Function	Renames a file or directory
os.walk()	Function	Generates file names in a directory tree

Predicting Heart Disease Using Machine Learning
Step :3

STEP 3: Check Kaggle Input Directory

what is os :

Common Use Cases:

Navigating and manipulating the file system

Manipulating file and directory paths

Managing environment variables

Running shell commands from Python scripts

Iterating over files and directories

Chapter 3

Predicting Heart Disease Using Machine Learning
Step :4

STEP 4: Load the Dataset

```
df = pd.read_csv("/kaggle/input/heart-disease-data/heart_disease_uci.csv")
df.head()
```

Explanation:

The CSV file is loaded into a pandas DataFrame named df.df.head() displays the first five rows.

Purpose: To confirm the dataset structure and view sample records

Chapter 3

Predicting Heart Disease Using Machine Learning
Step :4

STEP 4: Load the Dataset

	id	age	sex	dataset	cp	trestbps	chol	fbs	restecg	thalch	exang	oldpeak	slope	ca	thal	num
0	1	63	Male	Cleveland	typical angina	145.0	233.0	True	lv hypertrophy	150.0	False	2.3	downsloping	0.0	fixed defect	0
1	2	67	Male	Cleveland	asymptomatic	160.0	286.0	False	lv hypertrophy	108.0	True	1.5	flat	3.0	normal	2
2	3	67	Male	Cleveland	asymptomatic	120.0	229.0	False	lv hypertrophy	129.0	True	2.6	flat	2.0	reversable defect	1
3	4	37	Male	Cleveland	non-anginal	130.0	250.0	False	normal	187.0	False	3.5	downsloping	0.0	normal	0
4	5	41	Female	Cleveland	atypical angina	130.0	204.0	False	lv hypertrophy	172.0	False	1.4	upsloping	0.0	normal	0

Predicting Heart Disease Using Machine Learning
Step :5

STEP 5: Dataset Structure Information

Explanation:

Displays:

```
df.info()
```

Column names

Data types

Number of non-null values

Purpose: To understand which columns contain missing values and data types.

Chapter 3

Predicting Heart Disease Using Machine Learning

Step :5

STEP 5: Dataset Structure Information

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 920 entries, 0 to 919
Data columns (total 16 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   id        920 non-null    int64
 1   age       920 non-null    int64
 2   sex       920 non-null    object
 3   dataset   920 non-null    object
 4   cp        920 non-null    object
 5   trestbps  861 non-null    float64
 6   chol      890 non-null    float64
 7   fbs       830 non-null    object
 8   restecg   918 non-null    object
 9   thalch    865 non-null    float64
 10  exang     865 non-null    object
 11  oldpeak   858 non-null    float64
 12  slope     611 non-null    object
 13  ca        309 non-null    float64
 14  thal      434 non-null    object
 15  num       920 non-null    int64
dtypes: float64(5), int64(3), object(8)
memory usage: 115.1+ KB
```

Predicting Heart Disease Using Machine Learning
Step :6

STEP 6: Statistical Summary

```
df.describe()
```

Explanation:

Provides descriptive statistics for numerical columns:

Mean

Standard deviation

Minimum and maximum

Purpose: To understand data distribution and detect anomalies.

Chapter 3

Predicting Heart Disease Using Machine Learning

Step :6

STEP 6: Statistical Summary

	id	age	trestbps	chol	thalch	oldpeak	ca	num
count	920.000000	920.000000	861.000000	890.000000	865.000000	858.000000	309.000000	920.000000
mean	460.500000	53.510870	132.132404	199.130337	137.545665	0.878788	0.676375	0.995652
std	265.725422	9.424685	19.066070	110.780810	25.926276	1.091226	0.935653	1.142693
min	1.000000	28.000000	0.000000	0.000000	60.000000	-2.600000	0.000000	0.000000
25%	230.750000	47.000000	120.000000	175.000000	120.000000	0.000000	0.000000	0.000000
50%	460.500000	54.000000	130.000000	223.000000	140.000000	0.500000	0.000000	1.000000
75%	690.250000	60.000000	140.000000	268.000000	157.000000	1.500000	1.000000	2.000000
max	920.000000	77.000000	200.000000	603.000000	202.000000	6.200000	3.000000	4.000000

Predicting Heart Disease Using Machine Learning

Step :7

STEP 7: Categorical Summary

```
df.describe(include="O")
```

Explanation:

Shows statistics for categorical (object) columns.

Includes count, unique values, and most frequent values.

Purpose: To analyze non-numerical attributes.

Chapter 3

Predicting Heart Disease Using Machine Learning
Step :7

STEP 7: Categorical Summary

	sex	dataset	cp	fbs	restecg	exang	slope	thal
count	920	920	920	830	918	865	611	434
unique	2	4	4	2	3	2	3	3
top	Male	Cleveland	asymptomatic	False	normal	False	flat	normal
freq	726	304	496	692	551	528	345	196

Predicting Heart Disease Using Machine Learning

Step :8

STEP 8: Missing Values and Duplicates

```
df.isnull().sum()
df.duplicated().sum()
```

Explanation:

isnull().sum() counts missing values per column.

duplicated().sum() counts duplicate rows.

Purpose: To assess data quality before preprocessing.

Chapter 3

Predicting Heart Disease Using Machine Learning
Step :8

STEP 8: Missing Values and Duplicates

	0
id	0
age	0
sex	0
dataset	0
cp	0
trestbps	59
chol	30
fbs	90
restecg	2
thalch	55
exang	55
oldpeak	62
slope	309
ca	611
thal	486
num	0

dtype: int64

Predicting Heart Disease Using Machine Learning
Step :9

STEP 9: Handling Missing Values (Numerical)

```
df['trestbps'] = df['trestbps'].interpolate(method='linear')
df['chol'] = df['chol'].interpolate(method='linear')
```

Explanation:

Missing values are filled using linear interpolation.

This estimates values based on neighboring data.

Purpose:

To preserve numerical trends without deleting data.

Predicting Heart Disease Using Machine Learning

Step :10

STEP 10: Filling Missing Values Using Median

```
df['thalch'] = df['thalch'].fillna(df['thalch'].median())
df['oldpeak'] = df['oldpeak'].fillna(df['oldpeak'].median())
```

Explanation:

Median is used because it is robust to outliers.

Purpose:

To safely fill missing values without skewing data.

Predicting Heart Disease Using Machine Learning

Step :11

STEP 11: Rule-Based Imputation

```
df['fbs'] = df['fbs'].fillna(df.apply(
    lambda row: 1 if row['age'] > 50 else 0, axis=1))
```

Explanation:

If age > 50 → fasting blood sugar likely high.Uses domain knowledge.

Purpose:

To intelligently fill missing categorical values.

Predicting Heart Disease Using Machine Learning

Step :12

STEP 12: More Rule-Based Filling

```
df['exang'] = df['exang'].fillna(
    df.apply(lambda row: 1 if row['thalch'] < 120 else 0, axis=1))
```

Explanation:

Lower heart rate suggests exercise-induced angina

Purpose:

To improve data accuracy using medical logic.

Predicting Heart Disease Using Machine Learning

Step :13

STEP 13: Remaining Feature Imputation

```python
df['restecg'] = df['restecg'].fillna(
    df.apply(lambda row: 1 if row['exang']==1 else 0, axis=1))
```

```python
df['slope'] = df['slope'].fillna(
    df.apply(lambda row: 2 if row['oldpeak'] > 1 else 1, axis=1))
```

```python
df['thal'] = df['thal'].fillna(
    df.apply(lambda row: 7 if row['ca']>0 or row['slope']==2 else 3, axis=1))
```

```python
df['ca'] = df['ca'].fillna(
    df.apply(lambda row: 1 if row['thal']==3 or row['exang']==1 else 0, axis=1))
```

Predicting Heart Disease Using Machine Learning
Step :13

STEP 13: Remaining Feature Imputation

Explanation:

Missing values are filled using logical conditions based on related features.

Purpose:

To maintain medical consistency across variables.

Predicting Heart Disease Using Machine Learning
Step :14

STEP 14: Final Missing-Value Check

```
df.isnull().sum()
```

Explanation:

Confirms no missing values remain.

Predicting Heart Disease Using Machine Learning

Step :14

STEP 14: These two lines import visualization libraries used to create graphs and plots.

```
import matplotlib.pyplot as plt
import seaborn as sns
```

Explanation:

Matplotlib is a Python library used to create basic graphs

Importing the plotting module pyplot

Giving it the short name plt (standard convention)

Predicting Heart Disease Using Machine Learning

Step :14

STEP 14:

Matplotlib :TData visualization helps to understand complex datasets. The Matplotlib library in Python is a key tool for creating plots, and this guide walks you through installation and basic plotting.

Matplotlib is a popular plotting library in Python used for creating high-quality visualizations and graphs. It offers various tools to generate diverse plots, facilitating data analysis, exploration, and presentation. Matplotlib is flexible, supporting multiple plot types and customization options, making it valuable for scientific research, data analysis, and visual communication.

Predicting Heart Disease Using Machine Learning

Step :14

STEP 14:

Installation of Matplotlib:

Let's check how to set up the Matplotlib in Google Colab. Colab Notebooks are similar to Jupyter Notebooks except for running on the cloud. It is also connected to our Google Drive, making it much easier to access our Colab notebooks anytime, anywhere, and on any system. You can install Matplotlib by using the PIP command.

```
!pip install matplotlib
```

Predicting Heart Disease Using Machine Learning

Step :14

STEP 14:

To verify the installation, you would have to write the following code chunk:

```
import matplotlib
print(matplotlib.__version__)
```

different categories of plots that Matplotlib provides.

- Line plot
- Histogram
- Bar Chart
- Scatter plot
- Pie charts
- Boxplot

Predicting Heart Disease Using Machine Learning

Step :14

STEP 14:

Matplotlib is used to:

Create figures

Display plots

Set titles, labels, axes

Show graphs on screen

What is Seaborn:

Seaborn is a Python library built on top of Matplotlib

It provides: More beautiful plots and Statistical visualizations and Less code for complex graphs

Chapter 3

Predicting Heart Disease Using Machine Learning

Step :14

Seaborn is used to:

- Plot distributions
- Create boxplots, histograms, heatmaps
- Visualize relationships between variables

In this project(heart disease) these libraries are used to:

- Visualize age distribution
- Compare blood pressure vs blood sugar
- Show cholesterol vs heart rate
- Display correlation heatmap

These visualizations help:

- Understand data patterns
- Detect outliers
- Support modeling decisions

Predicting Heart Disease Using Machine Learning
Step :15

STEP 15: Data Visualization (EDA) Age Distribution

```
plt.figure()
```

plt.figure() creates a new, empty figure (plot window) where graphs will be drawn.

Why Do We Need plt.figure() :

Without plt.figure():

New plots may be drawn on top of old plots

Graphs may overlap and become unclear

With plt.figure():

Each plot is drawn separately

Graphs are clean and readable

Predicting Heart Disease Using Machine Learning
Step :15

STEP 15: Data Visualization (EDA) Age Distribution

```
plt.figure()
sns.histplot(df['age'], bins=20, kde=True)
plt.title("Distribution of Age")
plt.show()
```

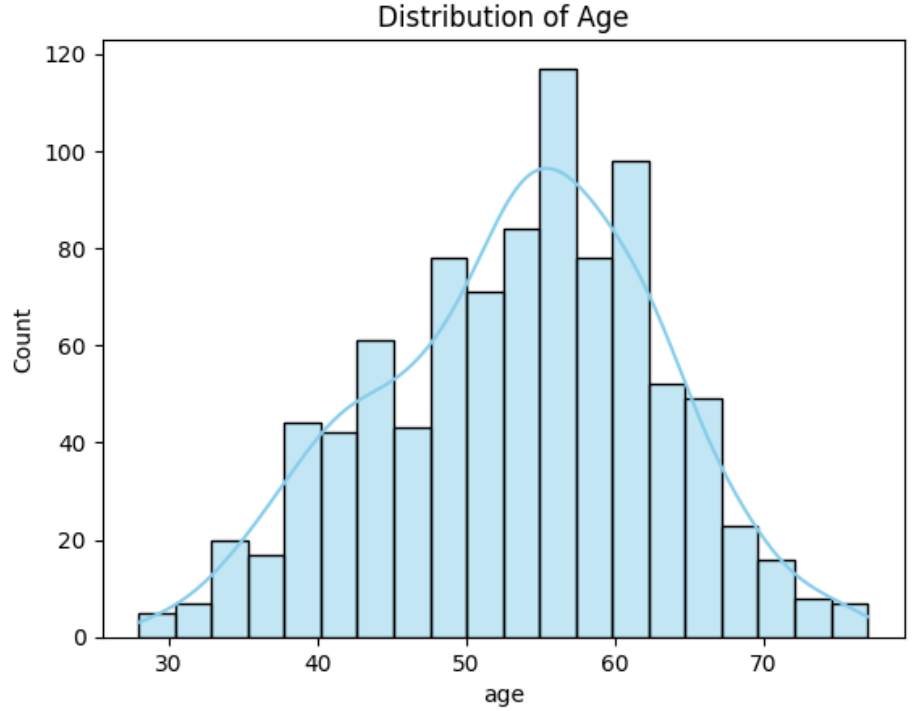
plt.figure() → opens a new figure
sns.histplot() → draws the histogram
plt.title() → adds a title
plt.show() → displays the figure

Chapter 3

Predicting Heart Disease Using Machine Learning
Step :15

STEP 15: Data Visualization (EDA) Age Distribution

This chart shows the distribution of patient ages in the heart disease dataset.
The histogram represents the frequency of patients within different age ranges, while the smooth curve (KDE) illustrates the overall age distribution trend.

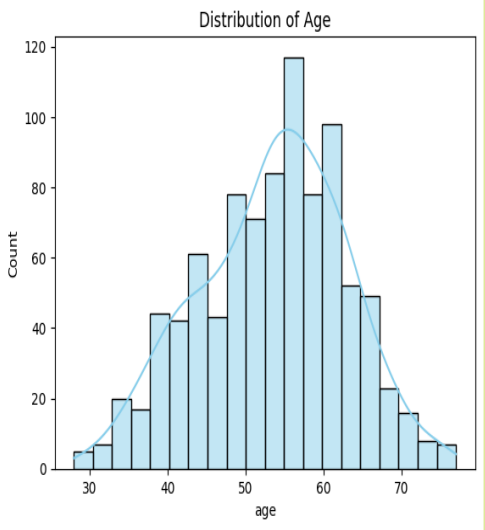


Chapter 3

Predicting Heart Disease Using Machine Learning

Step :15

STEP 15: Data Visualization (EDA) Age Distribution



Key observations:

Most patients are between 45 and 65 years old.

The distribution is approximately normal (bell-shaped).

Very few patients are younger than 30 or older than 75.

Relevance to classification:

Age is an important feature in heart disease prediction, as the likelihood of heart disease generally increases with age. Understanding the age distribution helps ensure that the classification model is trained on a representative population and can learn meaningful patterns related to age

Predicting Heart Disease Using Machine Learning
Step :15

STEP 15: Data Visualization (EDA)
Age Distribution

```
sns.histplot(df['age'], bins=20, kde=True)
```

Shows age distribution and density.

Predicting Heart Disease Using Machine Learning

Step :15

STEP 15: Data Visualization (EDA)
Blood Pressure vs FBS

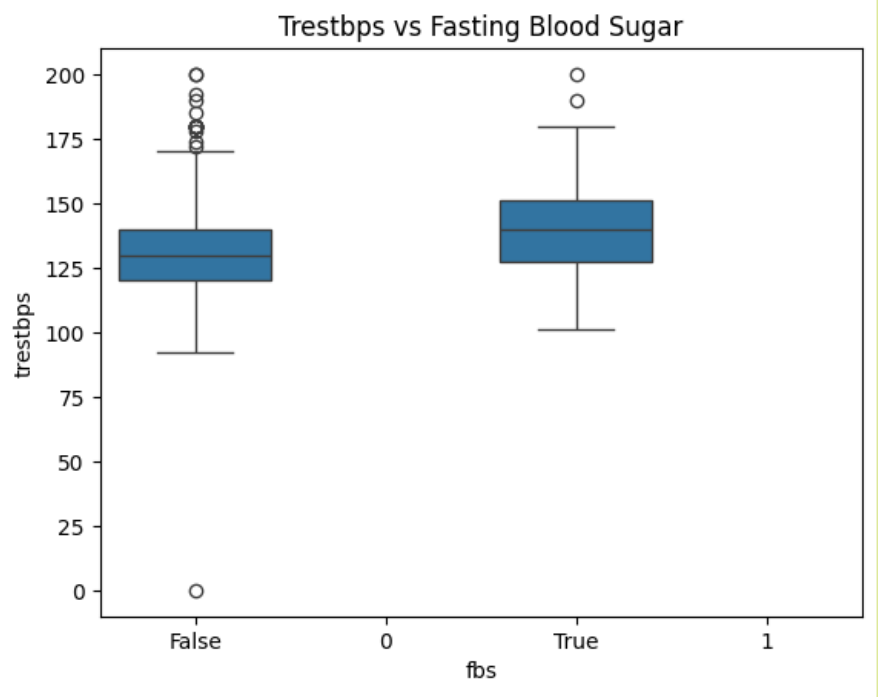
```
sns.boxplot(x='fbs', y='trestbps', data=df)
```

Compares blood pressure by sugar level.

Predicting Heart Disease Using Machine Learning
Step :15

STEP 15: Data Visualization (EDA)
Blood Pressure vs FBS

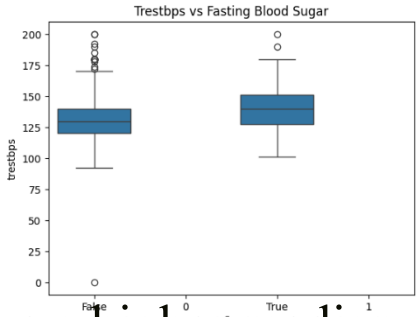
This box plot compares resting blood pressure (trestbps) between patients with normal fasting blood sugar and those with elevated fasting blood sugar (fbs).



Chapter 3

Predicting Heart Disease Using Machine Learning

Step :15



STEP 15: Data Visualization (EDA)

Blood Pressure vs FBS

Key observations:

Patients with high fasting blood sugar (fbs = True) tend to have a higher median resting blood pressure.

The blood pressure values for the high-FBS group show greater variability.

Outliers are present in both groups, indicating unusually high or low blood pressure readings.

Relevance to classification:

This visualization suggests a relationship between blood sugar levels and blood pressure, both of which are important risk factors for heart disease. Such patterns help the classification model learn how combinations of medical features contribute to the prediction of heart disease.

Predicting Heart Disease Using Machine Learning
Step :15

STEP 15: Data Visualization (EDA)
Cholesterol vs Heart Rate

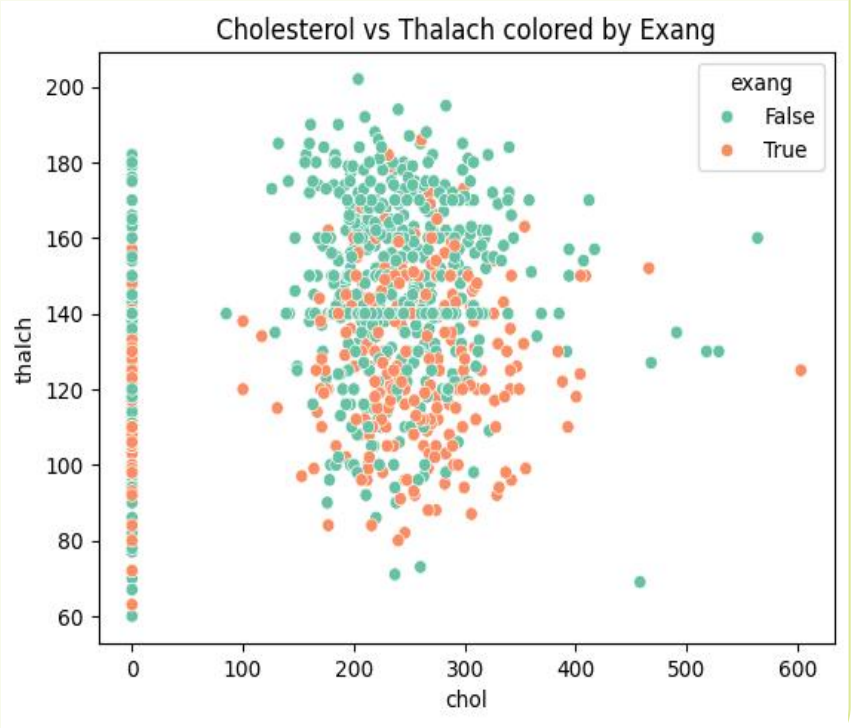
```
sns.scatterplot(x='chol', y='thalch', hue='exang', data=df)
```

Visualizes relationship between cholesterol and heart rate.

Predicting Heart Disease Using Machine Learning
Step :15

STEP 15: Data Visualization (EDA)

This scatter plot shows the relationship between cholesterol level (chol) and maximum heart rate achieved (thalach), with points colored based on the presence of exercise-induced angina (exang).



Predicting Heart Disease Using Machine Learning

Step :15

STEP 15: Data Visualization (EDA)

Key observations:

Patients without exercise-induced angina (exang = False) generally achieve higher maximum heart rates.

Patients with exercise-induced angina (exang = True) tend to have lower thalach values, even at similar cholesterol levels.

Cholesterol alone does not show a strong linear relationship with maximum heart rate, indicating that multiple features must be considered together.

Relevance to classification:

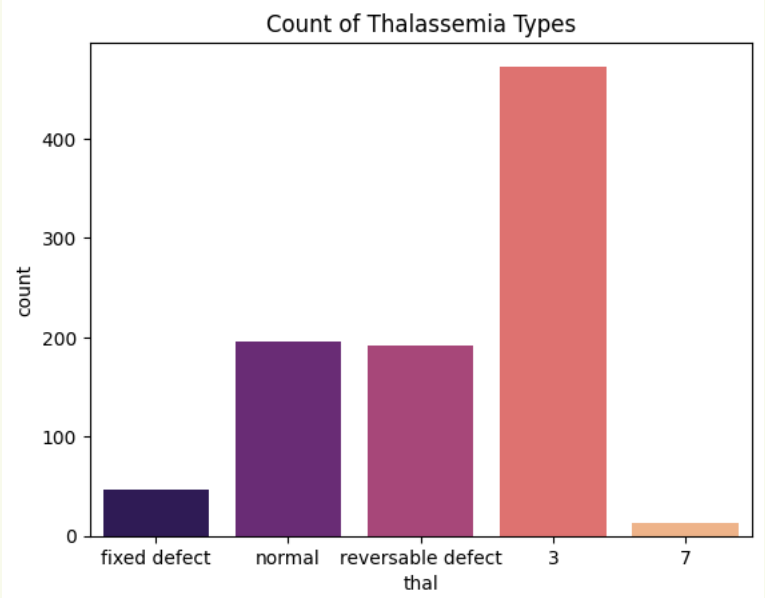
This visualization highlights how combinations of features (cholesterol, heart rate, and exercise-induced angina) help distinguish between different patient conditions. Such interactions are valuable for classification models like Random Forest, which can capture complex, non-linear relationships in medical data.

Chapter 3

Predicting Heart Disease Using Machine Learning
Step :15

STEP 15:

```
plt.figure()
sns.countplot(x='thal', palette='magma', data=df)
plt.title("Count of Thalassemia Types")
plt.show()
```



Predicting Heart Disease Using Machine Learning
Step :15

STEP 15:Count of Thalassemia Types (Count Plot)

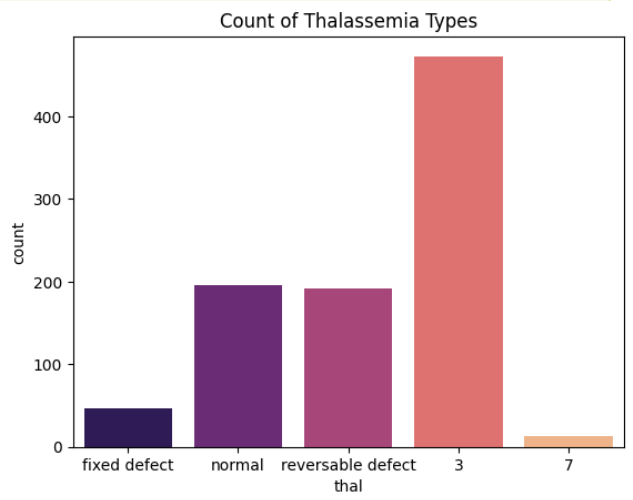
This chart displays the frequency distribution of thalassemia (thal) categories in the heart disease dataset.

Key observations:

The normal thalassemia category is the most frequent in the dataset.

Reversible defect and fixed defect cases appear less frequently.

A small number of cases fall into other categories, indicating class imbalance among thalassemia types.



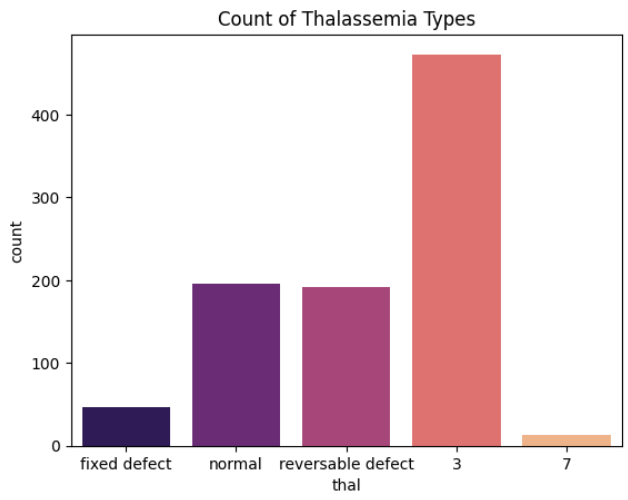
Chapter 3

Predicting Heart Disease Using Machine Learning
Step :15

STEP 15: Count of Thalassemia Types (Count Plot)

Relevance to classification:

Thalassemia type is an important clinical feature related to heart disease risk. Understanding its distribution helps the classification model learn how different thalassemia conditions contribute to heart disease prediction and highlights potential class imbalance that the model must handle.



Predicting Heart Disease Using Machine Learning

Step :15

STEP 15:

```
sns.barplot(x='fbs', y='chol', data=df, palette='Set3')
```

sns.barplot() is a Seaborn function used to create a bar chart that shows the relationship between two variables:

One categorical variable (x-axis)

One numerical variable (y-axis)

By default, it shows the mean (average) of the numerical variable for each category.

Predicting Heart Disease Using Machine Learning
Step :15

STEP 15:

```
sns.barplot(x='fbs', y='chol', data=df, palette='Set3')
```

x='fbs’

fbs = Fasting Blood SugarThis is a categorical variable:

0 → Normal fasting blood sugar

1 → High fasting blood sugar

These categories appear on the x-axis..

Predicting Heart Disease Using Machine Learning
Step :15

STEP 15:

```
sns.barplot(x='fbs', y='chol', data=df, palette='Set3')
```

y='chol'

chol = Serum cholesterol level

This is a numerical variable

The average cholesterol is plotted on the y-axis.

Predicting Heart Disease Using Machine Learning
Step :15

STEP 15:

```
sns.barplot(x='fbs', y='chol', data=df, palette='Set3')
```

data=df

df is the pandas DataFrame containing the dataset

Seaborn takes data directly from this table

No need to manually extract columns

Predicting Heart Disease Using Machine Learning

Step :15

STEP 15:

```
sns.barplot(x='fbs', y='chol', data=df, palette='Set3')
```

palette='Set3' :

Defines the color theme of the bars

Set3 is a predefined color palette in Seaborn

Improves visual clarity and aesthetics.

This bar plot shows:

The average cholesterol level for each fasting blood sugar group (normal vs high).

Each bar = one fbs category

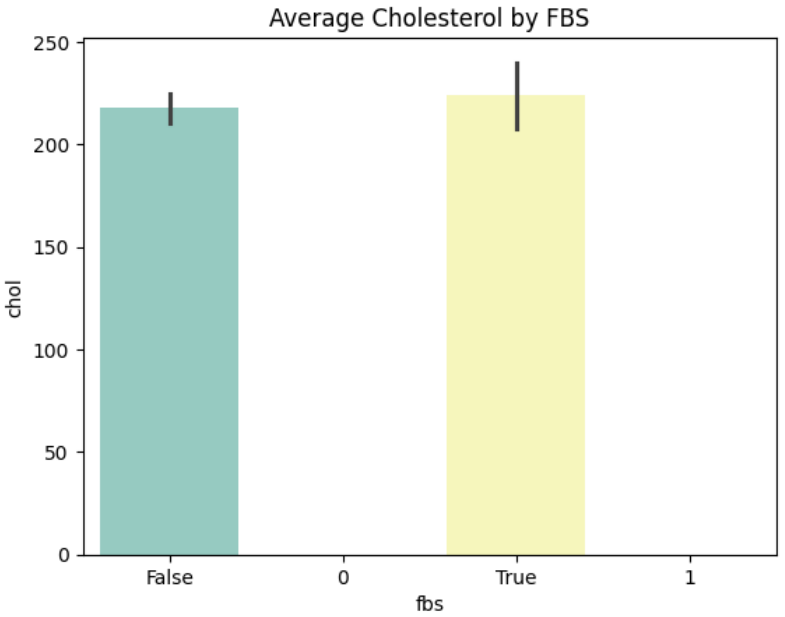
Height of bar = mean cholesterol value

Chapter 3

Predicting Heart Disease Using Machine Learning
Step :15

STEP 15:

This bar chart illustrates the average cholesterol level (chol) of patients grouped by their fasting blood sugar status (fbs).
The fbs variable indicates whether a patient's fasting blood sugar level is higher than the clinical threshold (typically >120 mg/dL).



Predicting Heart Disease Using Machine Learning
Step :15

STEP 15:

Description of the Chart:

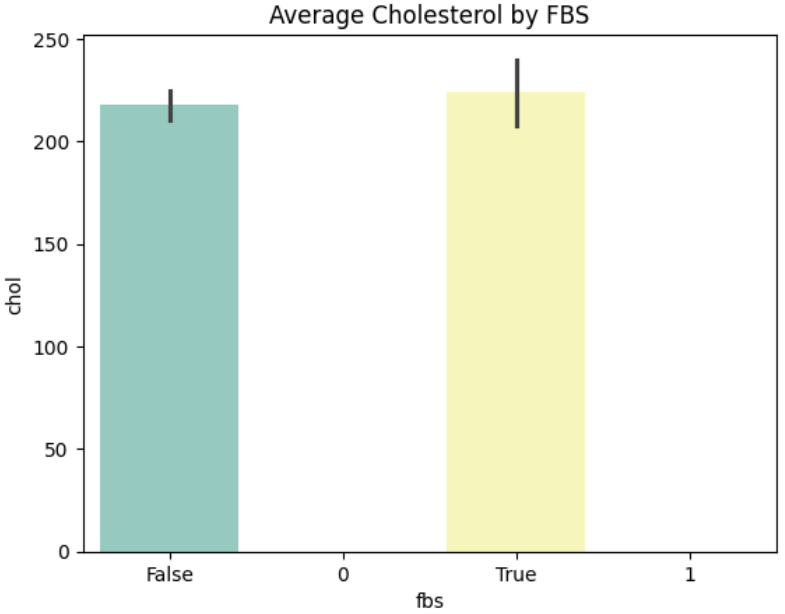
The x-axis represents fasting blood sugar categories:

False (0) → Normal fasting blood sugar

True (1) → Elevated fasting blood sugar

The y-axis shows the mean cholesterol level for each group.

The vertical black lines on each bar represent confidence intervals, indicating the variability of cholesterol values within each group.



Predicting Heart Disease Using Machine Learning
Step :15

STEP 15:

Key Observations:
Patients with elevated fasting blood sugar (fbs = True) tend to have a slightly higher average cholesterol level compared to those with normal fasting blood sugar.
Although the difference is not extremely large, it suggests a positive association between high blood sugar and higher cholesterol levels.
The overlap in confidence intervals indicates that cholesterol alone may not fully separate the two groups.

Predicting Heart Disease Using Machine Learning
Step :15

STEP 15:

Relevance to Classification:
This visualization highlights the relationship between fasting blood sugar and cholesterol, both of which are known cardiovascular risk factors. While neither feature alone may be sufficient to predict heart disease, their combined effect can improve the performance of a classification model such as Random Forest. This supports the use of multiple clinical features together to achieve more accurate heart disease prediction.

Predicting Heart Disease Using Machine Learning

Step :16

STEP 16: Boolean to Numeric Conversion

```
df['fbs'] = df['fbs'].map({True:1, False:0})
df['exang'] = df['exang'].map({True:1, False:0})
```

Explanation:

Converts Boolean values to numeric form for ML models.

Chapter 3

Predicting Heart Disease Using Machine Learning

Step :16

STEP 16:

```
sns.countplot(x='thal', hue='sex', data=df, palette='Set2')
```

countplot shows the number of observations (counts) for each category.

It is used for categorical data only

thal represents thalassemia / stress test result:

This is a categorical feature

Values represent different test outcomes

Appears on the x-axi

.

Predicting Heart Disease Using Machine Learning

Step :16

STEP 16:

```
sns.countplot(x='thal', hue='sex', data=df, palette='Set2')
```

hue='sex' :

sex divides the data into subgroups:

0 → Female

1 → Male

Bars are split by sex, shown in different colors

data=df:

df is the DataFrame containing the dataset

Predicting Heart Disease Using Machine Learning

Step :16

STEP 16:

```
sns.countplot(x='thal', hue='sex', data=df, palette='Set2')
```

palette='Set2':

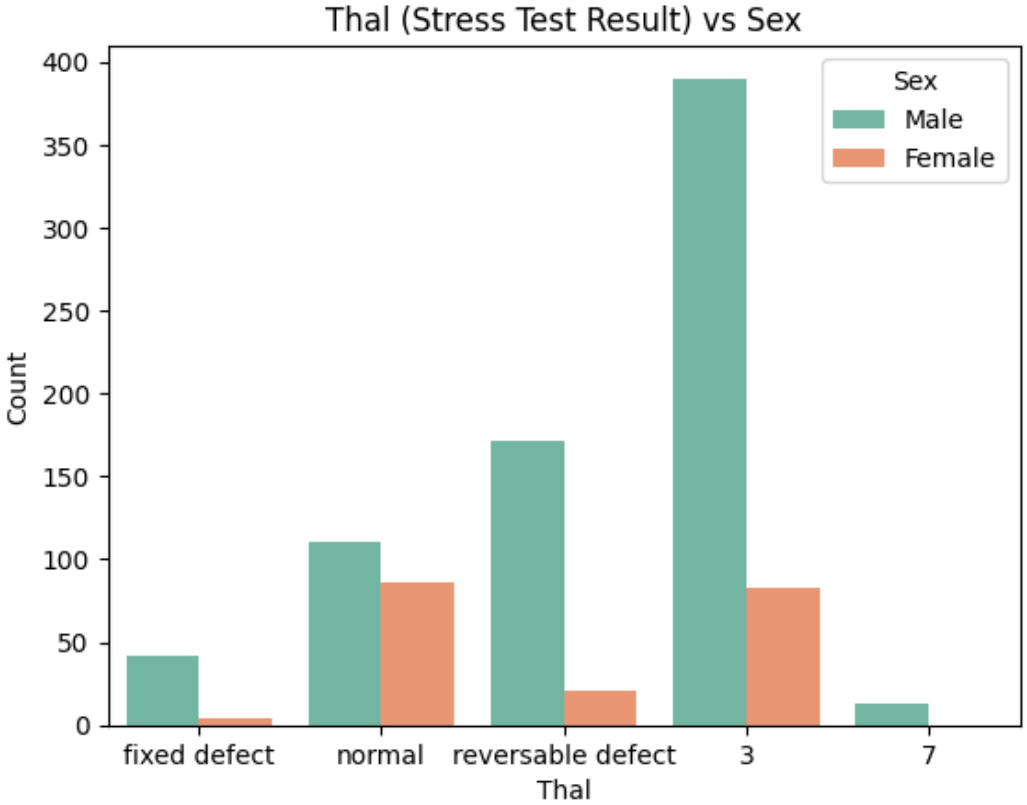
Specifies the color scheme

Improves readability and presentation quality

Predicting Heart Disease Using Machine Learning

Step :16

STEP 16:



Predicting Heart Disease Using Machine Learning

Step :16

STEP 16:

```
plt.title("Thal (Stress Test Result) vs Sex")
```

Adding a Title :

Adds a descriptive title to the graph

Clearly explains what the visualization represents

Titles are essential in academic plots

Predicting Heart Disease Using Machine Learning

Step :16

STEP 16:

```
plt.xlabel("Thal")
plt.ylabel("Count")
```

Labeling the Axes:

Explanation:

X-axis → Thal categories

Y-axis → Number of patients in each category

Axis labels improve interpretability.

Predicting Heart Disease Using Machine Learning

Step :16

STEP 16:

```
plt.legend(title="Sex")
```

Adding a Legend:

Explanation:

Displays a legend showing:

Male vs Female categories

Helps distinguish the colored bars

Predicting Heart Disease Using Machine Learning

Step :16

STEP 16:

```
plt.show()
```

Displaying the Plot :

Explanation:

Renders the plot on the screen

Without this line, the graph may not appear

Predicting Heart Disease Using Machine Learning
Step :17

STEP 17: Correlation Heatmap

```
sns.heatmap(df[['age','trestbps','chol','thalch','oldpeak','ca','fbs','exang']].corr(),
            annot=True, cmap='coolwarm')
```

Explanation:

Shows correlation between important numerical features.

Helps identify strong relationships.

Purpose:

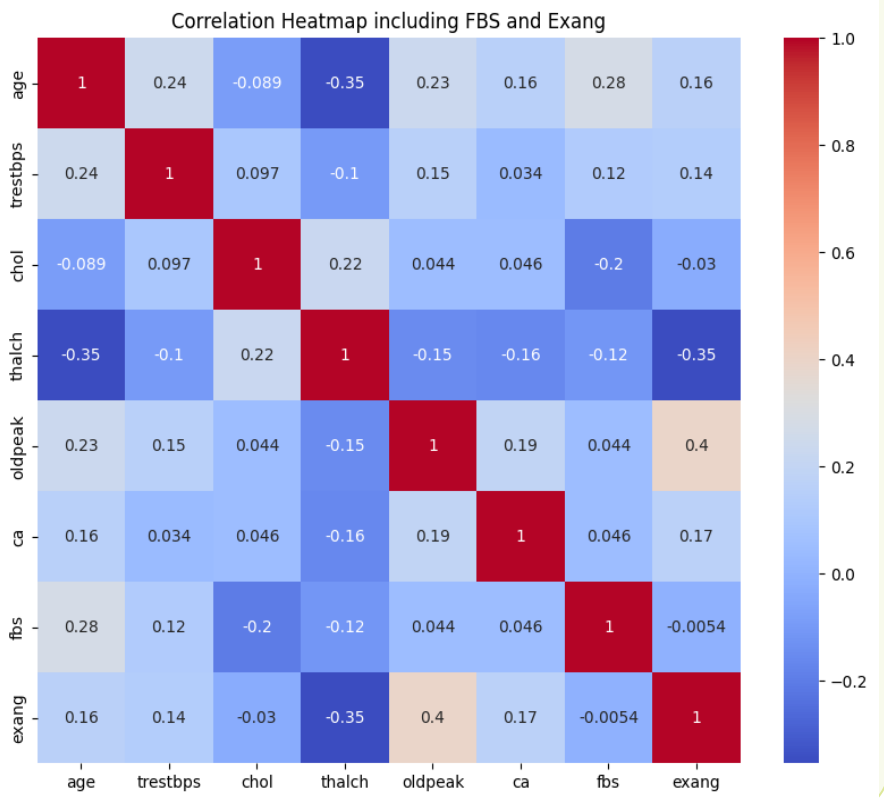
Feature understanding before classification

Chapter 3

Predicting Heart Disease Using Machine Learning
Step :17

STEP 17: Correlation Heatmap

.



Predicting Heart Disease Using Machine Learning

Step :18

STEP 18: STEP 18: Define Features and Target (Classification Setup)

```
import pandas as pd

df = pd.read_csv("/kaggle/input/heart-disease-data/heart_disease_uci.csv")
df = pd.get_dummies(df, drop_first=True)
# Target variable (heart disease)
y = df['num'].apply(lambda x: 1 if x > 0 else 0)

# Feature variables (all except target)
X = df.drop('num', axis=1)
```

What this considers:

y → class label (0 = no disease, 1 = disease)

X → medical features used for prediction

Predicting Heart Disease Using Machine Learning

Step :18

STEP 19: Train–Test Split

What this considers:

70% training data

30% testing data

stratify=y keeps class balance

```
from sklearn.model_selection import train_test_split


X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.30,
    random_state=42,
    stratify=y
)
```

Purpose: To evaluate the model on unseen data

Predicting Heart Disease Using Machine Learning
Step :20

STEP 20: Feature Scaling (Recommended)

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train.astype(float))
X_test = scaler.transform(X_test.astype(float))
```

What this considers:

Different feature scales (e.g., age vs cholesterol)

Improves model stability

Predicting Heart Disease Using Machine Learning

Step :21

STEP 21: Model Training (Random Forest Classifier)

What this considers:

Ensemble learning

Reduces overfitting

Suitable for medical classification

This is the actual classification model

```
from sklearn.ensemble import RandomForestClassifier


rf_model = RandomForestClassifier(
    n_estimators=100,
    random_state=42
)


rf_model.fit(X_train, y_train)
```

Predicting Heart Disease Using Machine Learning
Step :22

STEP 22: Prediction

```
y_pred = rf_model.predict(X_test)
```

What this does:

Predicts heart disease class for test patients

Predicting Heart Disease Using Machine Learning

Step :22

STEP 23: Model Evaluation

Accuracy:

```
from sklearn.metrics import accuracy_score

accuracy = accuracy_score(y_test, y_pred)
print("Model Accuracy:", accuracy)
```

Accuracy shows how many predictions were correct

Model Accuracy: 0.8840579710144928

Predicting Heart Disease Using Machine Learning

Step :22

STEP 23: Model Evaluation

Confusion Matrix:

```
from sklearn.metrics import confusion_matrix


cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", cm)
```

Explains:

True Positives-True Negatives-False Positives-False Negatives

Predicting Heart Disease Using Machine Learning

Step :22

STEP 23: Model Evaluation

Confusion Matrix:

Confusion Matrix: [[104 19]
[13 140]]

Predicting Heart Disease Using Machine Learning

Step :22

STEP 23: Model Evaluation

Classification Report (Recommended)

```
from sklearn.metrics import classification_report


print(classification_report(y_test, y_pred))
```

Includes:

Precision

Recall

F1-score

Predicting Heart Disease Using Machine Learning

Step :22

STEP 23: Model Evaluation

Classification Report (Recommended)

	precision	recall	f1-score	support
0	0.89	0.85	0.87	123
1	0.88	0.92	0.90	153
accuracy			0.88	276
macro avg	0.88	0.88	0.88	276
weighted avg	0.88	0.88	0.88	276



Chapter 4

Summery,conclusions and recommendations

Advantages:

Mining Based Methods are cost-effective and efficient

Helps in identifying criminal suspects

Helps in predicting the risk of diseases

Helps Banks and Financial Institutions to identify defaulters so that they may approve Cards, Loan, etc.

Disadvantages:

Privacy:

When the data is either are chances that a company may give some information about their customers to other vendors or use this information for their profit.

Accuracy Problem: Selection of Accurate model must be there in order to get the best accuracy and result.

APPLICATIONS:

Marketing and Retailing

Manufacturing

Telecommunication Industry

Intrusion Detection

Education System

Fraud Detection

Chapter 5

References

1.GeeksforGeeks. (n.d.). Classification in data mining.
https://www.geeksforgeeks.org/machine-learning/classification-in-data-mining/

2. Waskom, M. L. (2021). Seaborn: Statistical data visualization. Journal of Open Source Software, 6(60), 3021.
https://doi.org/10.21105/joss.03021

3. Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90–95.
https://doi.org/10.1109/MCSE.2007.55

4. Kaggle. (2024). Kaggle: Your machine learning and data science community.
https://www.kaggle.com

5. Redwan Karim Sony. (2020). Heart disease dataset [Data set]. Kaggle. https://www.kaggle.com/datasets/redwankarimsony/heart-disease-data

6. OpenAI. (2023). ChatGPT (GPT-4 version) [Large language model]. https://chat.openai.com/

7. NumPy Developers. (n.d.). What is NumPy? NumPy documentation. https://numpy.org/doc/stable/user/whatisnumpy.html

8. GeeksforGeeks. (2025, December 5). Pandas introduction. GeeksforGeeks. https://www.geeksforgeeks.org/pandas/introduction-to-pandas-in-python

9. Pozo Ramos, L. (2025, July 16). os | Python Standard Library (Reference). Real Python. https://realpython.com/ref/stdlib/os/

10. Agarwal, G. (2025, April 15). Introduction to Matplotlib using Python for beginners. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/10/introduction-to-matplotlib-using-python-for-beginners/ Analytics Vidhya