SUMMER TRAINING/INTERNSHIP

PROJECT REPORT

(Term June–July 2025)

HEART DISEASE RISK PREDICTION USING EXPLORATORY DATA ANALYSIS AND DASHBOARDS

Submitted by

Harleen Kaur
Registration Number: 12326083

Himanshu Yadav
Registration Number: 12325588

Saksham
Registration Number: 12325154

Gitesh Battania
Registration Number: 12325999

Anmol Gautam
Registration Number: 12326272

Course Code: CSE343 / CSE443

Under the Guidance of

Miss Sandeep Kaur

School of Computer Science and Engineering

Acknowledgement

I would like to express my sincere gratitude to all those who supported me throughout the course of this project.

First and foremost, I extend my heartfelt thanks to Miss Sandeep Kaur, for their constant guidance, encouragement, and valuable feedback during the entire training period. Their support was instrumental in shaping this project and enhancing my learning experience.

I am also grateful to the School of Computer Science and Engineering at Lovely Professional University for providing this opportunity and the necessary infrastructure to carry out this training.

Special thanks to my team members for their cooperation, dedication, and seamless collaboration. Each one played a crucial role in making this project a success.

Lastly, I would like to thank my family and friends for their constant motivation and moral support throughout this journey.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

Company Profile

This training project was conducted under the academic curriculum of the School of Computer Science and Engineering at Lovely Professional University. It was aimed at providing practical exposure to real-world data analysis in the healthcare domain, which is one of the most impactful and data-driven sectors globally.

The objective was not tied to any external corporate internship but rather designed to simulate an industry-level experience by encouraging the use of professional tools and methodologies like Python, Power BI, and Exploratory Data Analysis (EDA) techniques. The focus of the training was to equip students with data handling, statistical analysis, and visualization skills, particularly relevant in health informatics — a growing field combining healthcare and data science.

Overview of Training Domain

The domain of this training was Healthcare Analytics, a specialized area within data science that focuses on analyzing health-related data to generate actionable insights. Healthcare analytics plays a vital role in modern medical research and decision-making, especially when it comes to identifying patterns, risk factors, and early indicators of disease.

Another critical aspect of the training was dashboard creation, which bridges the gap between complex data and user-friendly communication. Using tools like Power BI, we transformed the results of our EDA into dynamic, interactive dashboards. These dashboards allow healthcare professionals or analysts to explore the data intuitively and make better-informed decisions.

Objective of the Project

The primary objective of this project was to perform a comprehensive exploratory data analysis (EDA) on a heart disease dataset to uncover significant patterns, relationships, and trends that could help in the early prediction and understanding of heart disease risk.

Heart disease continues to be one of the leading causes of death worldwide. Therefore, identifying critical health indicators and risk factors at an early stage is vital for prevention and timely medical intervention. This project aimed to contribute to that goal by leveraging data analytics and visualization techniques. To explore and analyse real-world patient data and identify variables that strongly correlate with the presence or absence of heart disease. To use statistical tools and visual methods to interpret the influence of features such as age, cholesterol, chest pain type, blood pressure, and maximum heart rate on heart disease risk.

Overall, the objective was not just to perform analysis, but to convert raw data into insights that can guide decision-making and promote awareness about cardiovascular health risks.

CHAPTER 2: TRAINING OVERVIEW

Tools & Technologies Used

In this project, a combination of programming languages, libraries, and business intelligence tools was used to carry out data analysis, visualization, and dashboard development. Each tool played a specific and crucial role in different phases of the project:

1. Python (Jupyter Notebook):
Python was the core programming language used for data analysis. It is widely adopted in the data science industry due to its simplicity, readability, and powerful ecosystem. All the preprocessing and exploratory data analysis (EDA) tasks were performed using Python in a Jupyter Notebook environment. Jupyter provided an interactive platform to write, visualize, and explain code outputs step-by-step.

2. Pandas:
Pandas is a Python library used for data manipulation and analysis. It allowed us to load, clean, and transform the dataset effectively. With functions like read_csv(), describe(), groupby(), and isnull().sum(), we were able to manage and explore the structure of the dataset quickly and efficiently.

3. NumPy:
NumPy was used for handling numerical operations and working with arrays. It supported statistical computations and assisted in generating numerical summaries. Although used more subtly than Pandas, it underpinned many of the calculations and data transformations done during preprocessing.

4. Matplotlib:
Matplotlib is a 2D plotting library used to create static, animated, and interactive visualizations in Python. It was used to generate basic charts such as histograms and line plots to observe trends in features like age, cholesterol levels, and heart rate.

5. Seaborn:
Seaborn is built on top of Matplotlib and was used extensively for advanced visualization. It provided beautiful, statistical plots like countplots, heatmaps, and boxplots. These visualizations helped reveal patterns, correlations, and distributions in the dataset, especially when comparing features against the target variable (presence of heart disease).

6. Power BI:
Power BI was used for building the interactive dashboard at the final stage of the project. After completing EDA in Python, the cleaned dataset was imported into Power BI. It was used to create visual representations like bar charts, pie charts, filters, and slicers. Power BI’s drag-and-drop interface enabled the team to design a user-friendly and interactive dashboard that summarizes the project findings in a clear and dynamic manner.

Areas Covered During Training

- Data Cleaning

This involved identifying and handling missing values, correcting data types, removing duplicates, and encoding categorical variables. It ensured the dataset was consistent, reliable, and ready for analysis.

- Exploratory Data Analysis (EDA) and Machine Learning.

We performed univariate and bivariate analysis to understand the distribution and interaction of features. Visualization tools like Seaborn and Matplotlib were used to uncover hidden patterns and trends.

- Correlation and Statistical Understanding

Statistical summaries and correlation matrices were generated to examine the relationships between variables. This helped identify key predictors for heart disease, such as chest pain type and ST depression.

- Dashboard Creation using Power BI

The insights from EDA were presented through an interactive dashboard. Power BI was used to create visual summaries, filters, and charts that allow users to explore the dataset in an intuitive way.

Daily/Weekly Work Summary

Week 1 – Dataset Understanding and Preprocessing:

The first week was dedicated to exploring the dataset, understanding its structure, identifying key variables, and performing essential data cleaning tasks. This included handling missing values, removing duplicates, and formatting the data for analysis.

Week 2 – Exploratory Data Analysis (EDA):

During the second week, we performed univariate and bivariate analysis using Python libraries like Pandas, Seaborn, and Matplotlib. We created various plots to visualize distributions and relationships between variables and the target (heart disease).

Week 3 – Correlation Study and Insights:

We focused on statistical analysis such as calculating correlations, drawing heatmaps, and identifying the most influential features for heart disease prediction. This helped us prioritize variables for dashboard design.

Week 4 – Dashboard Creation and Finalization:

The last phase was dedicated to importing the cleaned dataset into Power BI and building the dashboard. Visuals were designed based on EDA insights, filters were added for interactivity, and the layout was refined for clarity. We also compiled the final report and prepared for the viva presentation.

CHAPTER 3: PROJECT DETAILS

Title of the Project

Heart Disease Risk Prediction using EDA and Dashboards

Problem Definition

Understanding the key factors contributing to heart disease and enabling data-driven risk prediction using visualization and analysis. Heart disease is a leading cause of death globally, and its prevention or early detection remains a critical challenge. While various health parameters such as age, cholesterol, blood pressure, and chest pain are known to influence cardiovascular health, making sense of these variables collectively requires advanced data analysis.

The problem this project addresses is the lack of intuitive, data-driven methods for identifying patterns that indicate higher heart disease risk. Medical data is often complex and difficult to interpret without specialized tools. By performing detailed EDA and converting insights into visual dashboards, this project aims to provide a solution that makes healthcare data more interpretable and actionable. It helps bridge the gap between raw data and decision-making, supporting early diagnosis and potentially saving lives through better understanding of contributing risk factors.

Scope and Objectives

The scope of this project encompasses the complete process of data analysis, from preprocessing raw medical data to transforming insights into visually interactive dashboards. The project specifically deals with heart disease-related patient data, which includes features such as age, gender, chest pain type, cholesterol levels, blood pressure, and more.

The key objectives of this project include:

- To explore the dataset through statistical analysis and understand the distribution of variables related to heart disease.
- To identify significant patterns and correlations between the target variable (presence of heart disease) and other features.
- To apply visualization techniques to effectively communicate findings to both technical and non-technical audiences.
- To build an interactive dashboard using Power BI that allows stakeholders (healthcare professionals, students, or analysts) to explore the data and insights in real time.

The overall objective is to simulate a real-world healthcare analytics workflow and present findings that can assist in early risk assessment of heart disease.

System Requirements

To successfully carry out this project, the following software and hardware requirements were necessary:

- Python 3.x – Programming language used for data cleaning, analysis, and visualization.
- Jupyter Notebook – Interactive development environment for writing and executing Python code.
- Pandas, NumPy, Matplotlib, Seaborn – Python libraries used for data handling and EDA, Machine Learning.
- Power BI (Desktop Version) – For creating dashboards and visual summaries of the insights.
- MS Excel – Optional support tool for data formatting and quick previews.
- Hardware Configuration – A system with at least:
    - 8GB RAM (recommended for smooth execution of Python scripts and Power BI)
    - Windows 10 or higher operating system
    - At least 2 GB of free disk space for handling large datasets and application installations

These tools combined to provide a robust environment for conducting end-to-end healthcare data analysis and visualization.

Architecture Diagram

The architecture of this project is structured around a sequential data analytics pipeline, consisting of the following key stages:

1. Data Collection:
The project began with the acquisition of a publicly available heart disease dataset. This dataset contained essential patient health attributes such as age, sex, chest pain type, cholesterol, blood pressure, and target (heart disease presence or absence).

2. Data Cleaning:
Raw data was preprocessed to ensure consistency and reliability. This step included handling missing values, removing duplicates, converting data types, and encoding categorical variables for analysis.

3. Exploratory Data Analysis (EDA):
Using Python libraries, various statistical summaries and visual plots were generated to

identify distributions, relationships, and potential indicators of heart disease. This step laid the foundation for selecting important features.

4. Visualization:
Meaningful trends and insights uncovered during EDA were visualized using tools like Matplotlib and Seaborn. These included count plots, histograms, heatmaps, and boxplots that revealed feature-to-feature and feature-to-target interactions.

5. Dashboard Creation:
The final cleaned and processed dataset was imported into Power BI, where the findings were organized into an interactive dashboard. Filters and slicers were added to allow users to view data segmented by age, gender, and other key attributes.

Workflow Summary:
Data Collection → Data Cleaning → EDA → Visualization → Dashboard Creation

This linear and structured workflow ensured a smooth and logical progression from raw data to decision-support visuals.

Data flow / UML Diagrams

For this project, UML or detailed system architecture diagrams were not applicable, as the focus was primarily on data exploration and visualization, rather than developing a full-fledged software application.

Instead of object-oriented or system design concepts, the project followed a data-centric approach, where the goal was to derive insights from structured health data and present them through clear and interactive visuals. The main deliverables were analytical findings and a Power BI dashboard—not software components requiring class diagrams, sequence diagrams, or data flow modeling.

However, the project structure and process flow were carefully documented and logically followed, similar to how data analytics teams operate in real-world industry environments.

CHAPTER 4: IMPLEMENTATION

Tools Used

The project leveraged a combination of data science libraries and a business intelligence platform to perform end-to-end analysis and visualization of the heart disease dataset.

- Python (Jupyter Notebook): The primary programming environment used for writing, testing, and running analysis code. It provided an interactive and iterative workflow ideal for data exploration.
- Pandas: A powerful Python library used for data manipulation and preprocessing. It allowed efficient loading, cleaning, transformation, and aggregation of the dataset.
- Seaborn: This library was used to generate advanced statistical visualizations such as count plots, box plots, and heatmaps. Seaborn helped reveal patterns and correlations between features.
- Matplotlib: Used alongside Seaborn for plotting basic charts and customizing visuals. It supported line plots, histograms, and bar graphs to supplement the exploratory phase.
- Power BI: A business intelligence tool used in the final stage of the project. It enabled the creation of an interactive and user-friendly dashboard that presents insights discovered during EDA. Filters and slicers were added for enhanced exploration.

These tools worked together to provide a complete data analytics and visualization pipeline from raw dataset to polished dashboard.

Methodology

The project followed a structured, step-by-step methodology to transition from raw data to insight generation and final presentation:

1. Loaded and Cleaned the Dataset
The dataset was imported using Pandas and underwent thorough preprocessing. This included checking for null values, handling data types, encoding categorical features, and removing duplicates to ensure data quality.

2. Explored Distributions and Relationships Between Variables
Using Pandas, Seaborn, and Matplotlib, we performed both univariate (single variable) and bivariate (two-variable) analysis. This helped us understand how features like age, chest pain type, cholesterol, and heart rate are distributed and related to the risk of heart disease.

3. Visualized Trends with Plots
Statistical insights were visualized using heatmaps, count plots, box plots, and bar charts. These visuals helped identify patterns, such as which variables had strong associations with heart disease.

4. Built a Dashboard in Power BI
After EDA, the cleaned and analyzed dataset was imported into Power BI. There, an interactive dashboard was created featuring multiple visualizations and slicers for filtering data by age group, gender, chest pain type, etc. The dashboard made the analysis accessible to non-technical users and stakeholders.

Modules / Screenshots

The entire project was structured in a modular format, with each component focusing on a specific stage of the data analytics lifecycle. Below is a breakdown of the core modules and what each achieved:

1. Data Cleaning Module
This module handled all preprocessing tasks:

- Checked for and removed missing or null values.
- Removed duplicate records to ensure data uniqueness.
- Encoded categorical variables (e.g., chest pain type, thalassemia) into numerical form for analysis.
- Standardized column names for better readability and consistency.

2. EDA Module (Univariate & Bivariate Analysis)
Here, data exploration was carried out using:

- Count plots for categorical variables like sex, chest pain type, and fasting blood sugar.
- Histograms for numerical features like age, cholesterol, and maximum heart rate.
- Box plots to observe spread, central tendencies, and outliers in key features.

3. Correlation Analysis Module

- Generated a correlation matrix to observe relationships between features and the target variable.
- Used a heatmap (via Seaborn) to visually represent how strongly each feature correlates with heart disease.
- Identified significant predictors like chest pain type, ST depression (oldpeak), and max heart rate.

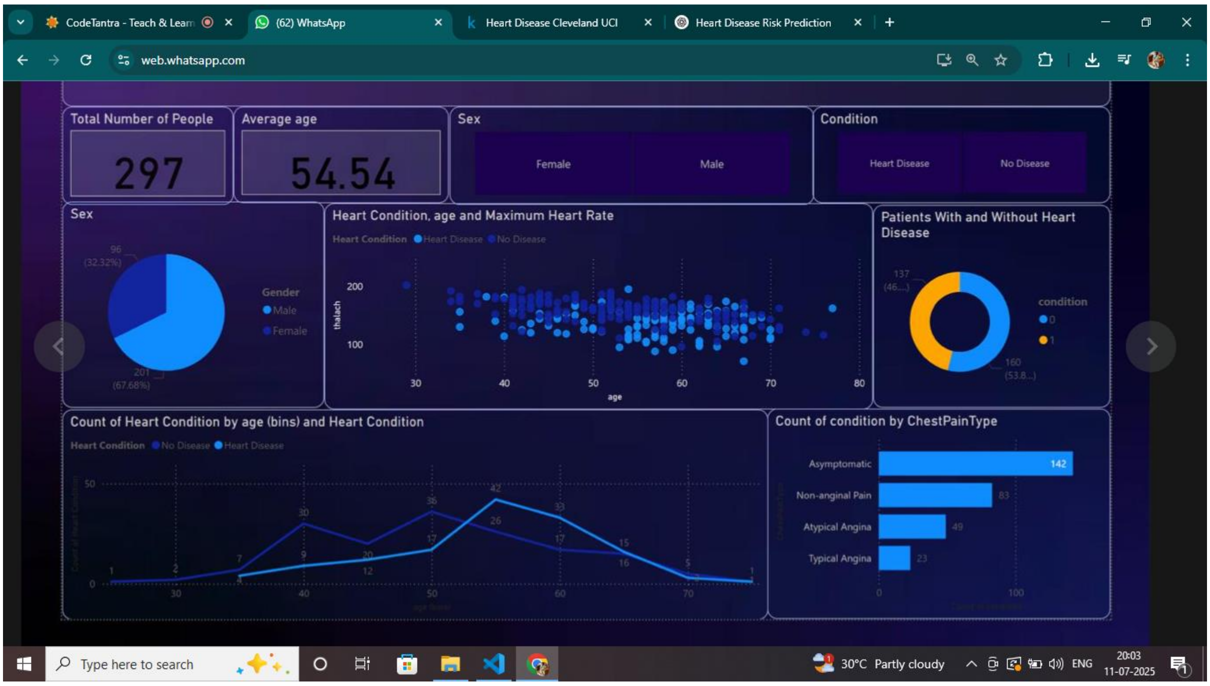
4. Power BI Dashboard Module

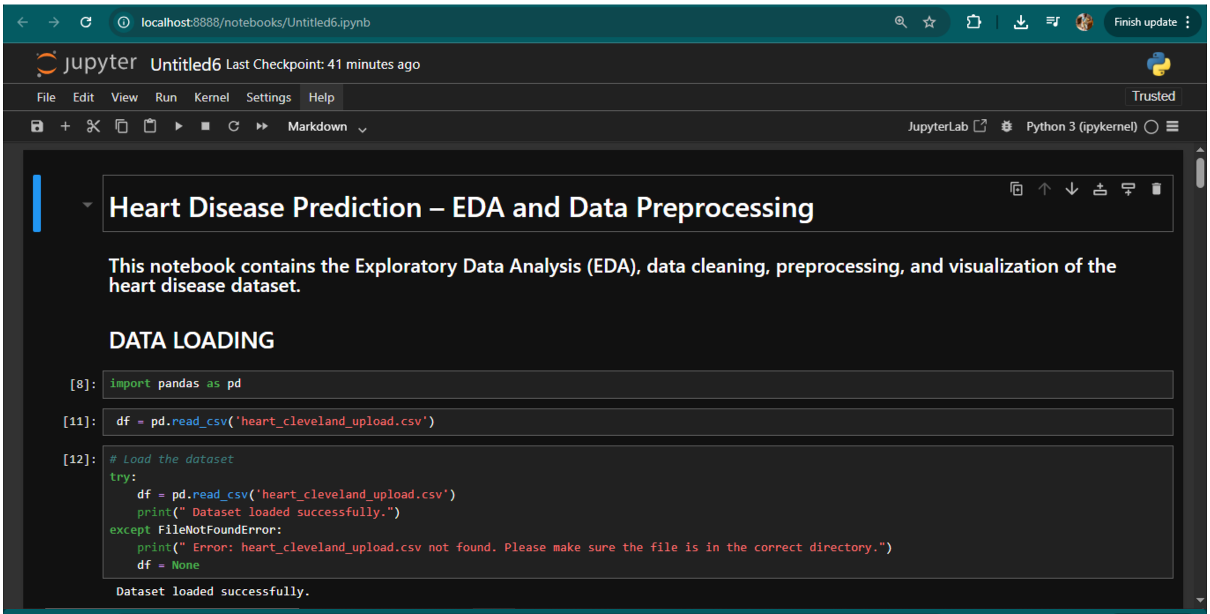
- Imported the cleaned dataset into Power BI.
- Created bar charts, pie charts, card visuals (showing totals), and filters for user interaction.
- Integrated slicers for filtering data by gender, age, and chest pain type to make the insights dynamic and customizable.

Code snippets

Key code used included:
- df.describe(), df.isnull().sum()
- sns.heatmap(), sns.countplot(), sns.boxplot()
- Correlation matrices and groupby visual summaries

Screenshots:




localhost:8888/notebooks/Untitled6.ipynb
Finish update
jupyter Untitled6 Last Checkpoint: 41 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Markdown
JupyterLab
Python 3 (ipykernel)

Heart Disease Prediction – EDA and Data Preprocessing

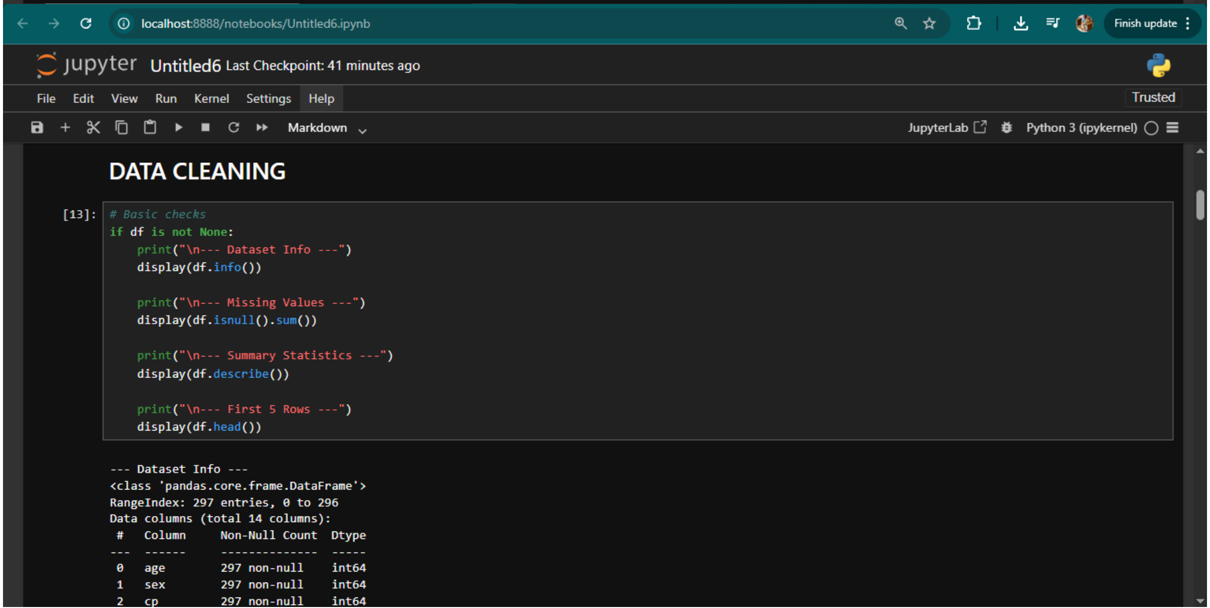
This notebook contains the Exploratory Data Analysis (EDA), data cleaning, preprocessing, and visualization of the heart disease dataset.

DATA LOADING

```
[8]: import pandas as pd
```

```
[11]: df = pd.read_csv('heart_cleveland_upload.csv')
```

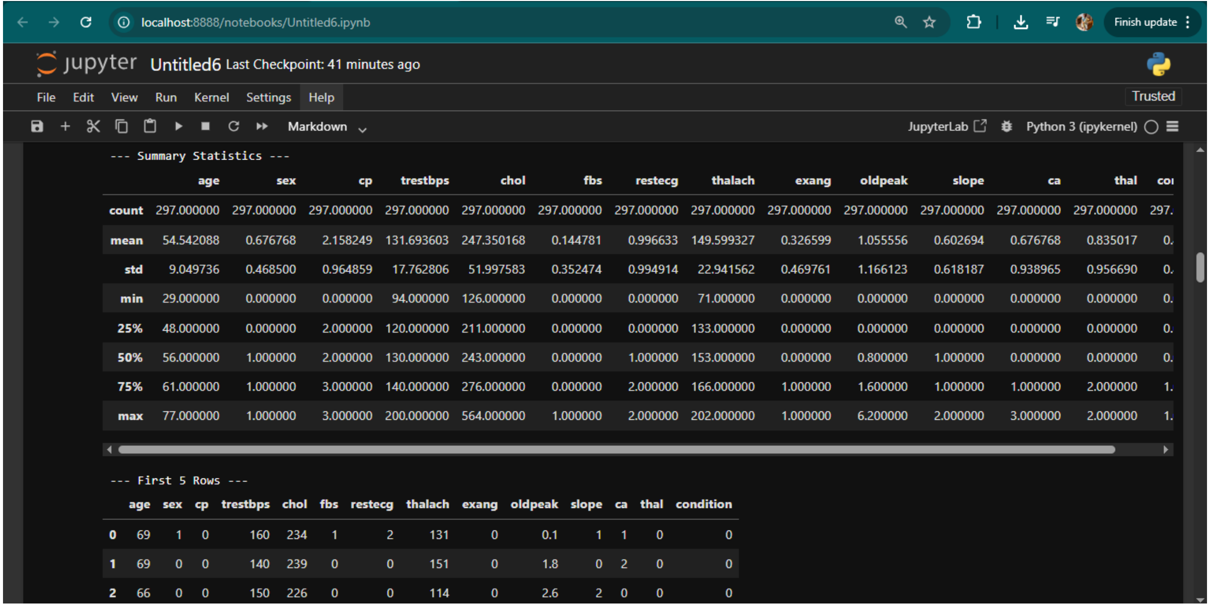
```
[12]: # Load the dataset
try:
    df = pd.read_csv('heart_cleveland_upload.csv')
    print(" Dataset loaded successfully.")
except FileNotFoundError:
    print(" Error: heart_cleveland_upload.csv not found. Please make sure the file is in the correct directory.")
    df = None

Dataset loaded successfully.
```


localhost:8888/notebooks/Untitled6.ipynb
Finish update
jupyter Untitled6 Last Checkpoint: 41 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Markdown
JupyterLab
Python 3 (ipykernel)

DATA CLEANING

```
[13]: # Basic checks
if df is not None:
    print("\n--- Dataset Info ---")
    display(df.info())

    print("\n--- Missing Values ---")
    display(df.isnull().sum())

    print("\n--- Summary Statistics ---")
    display(df.describe())

    print("\n--- First 5 Rows ---")
    display(df.head())
```

```
--- Dataset Info ---
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 297 entries, 0 to 296
Data columns (total 14 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   age        297 non-null    int64
 1   sex        297 non-null    int64
 2   cp         297 non-null    int64
```

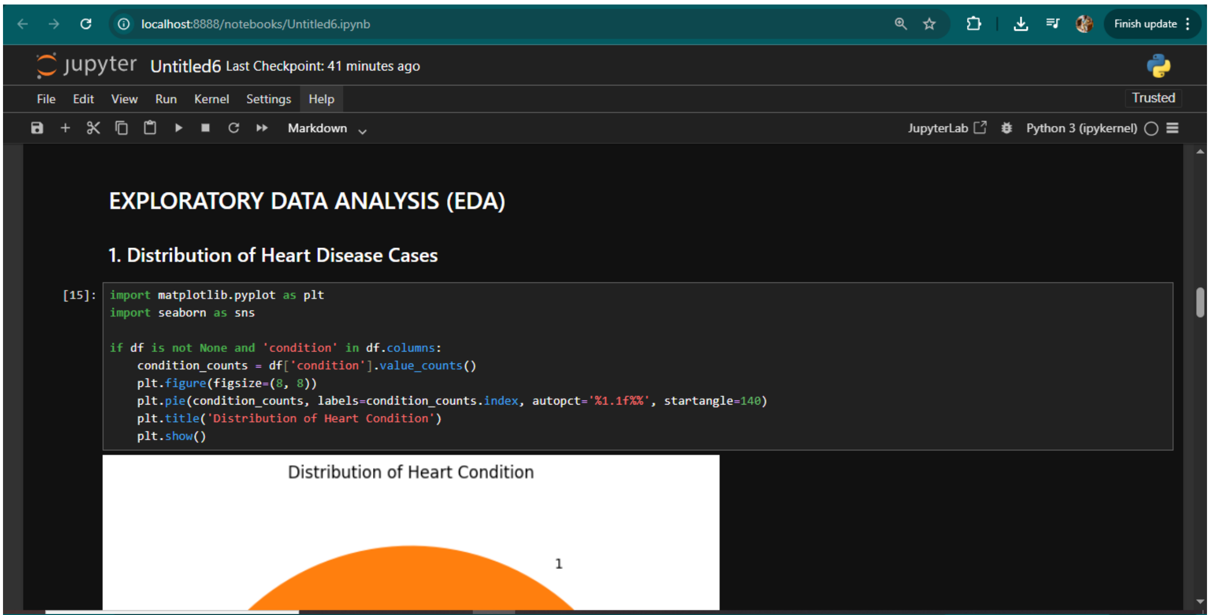

localhost:8888/notebooks/Untitled6.ipynb
Finish update
jupyter Untitled6 Last Checkpoint: 41 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Markdown
JupyterLab
Python 3 (ipykernel)

```
--- Summary Statistics ---
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	co
count	297.000000	297.000000	297.000000	297.000000	297.000000	297.000000	297.000000	297.000000	297.000000	297.000000	297.000000	297.000000	297.000000	297.
mean	54.542088	0.676768	2.158249	131.693603	247.350168	0.144781	0.996633	149.599327	0.326599	1.055556	0.602694	0.676768	0.835017	0.
std	9.049736	0.468500	0.964859	17.762806	51.997583	0.352474	0.994914	22.941562	0.469761	1.166123	0.618187	0.938965	0.956690	0.
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.
25%	48.000000	0.000000	2.000000	120.000000	211.000000	0.000000	0.000000	133.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.
50%	56.000000	1.000000	2.000000	130.000000	243.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	0.000000	0.
75%	61.000000	1.000000	3.000000	140.000000	276.000000	0.000000	2.000000	166.000000	1.000000	1.600000	1.000000	1.000000	2.000000	1.
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	3.000000	2.000000	1.

```
--- First 5 Rows ---
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	condition
0	69	1	0	160	234	1	2	131	0	0.1	1	1	0	0
1	69	0	0	140	239	0	0	151	0	1.8	0	2	0	0
2	66	0	0	150	226	0	0	114	0	2.6	2	0	0	0

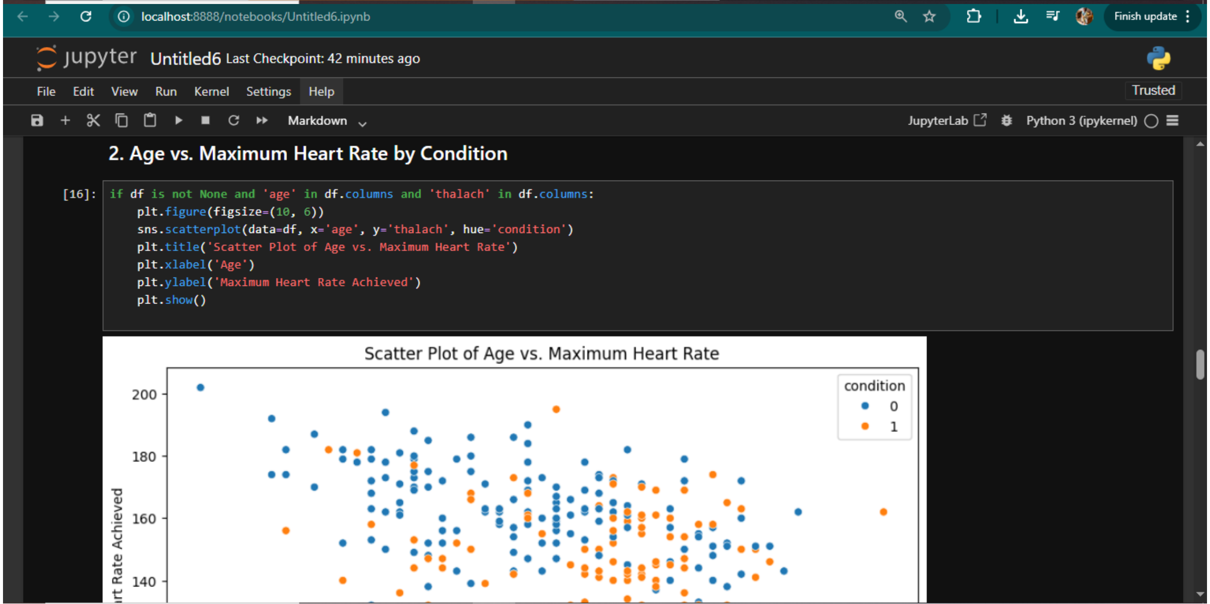

localhost:8888/notebooks/Untitled6.ipynb
Finish update
jupyter Untitled6 Last Checkpoint: 41 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Markdown
JupyterLab
Python 3 (ipykernel)

EXPLORATORY DATA ANALYSIS (EDA)

1. Distribution of Heart Disease Cases

[15]:
```
import matplotlib.pyplot as plt
import seaborn as sns

if df is not None and 'condition' in df.columns:
    condition_counts = df['condition'].value_counts()
    plt.figure(figsize=(8, 8))
    plt.pie(condition_counts, labels=condition_counts.index, autopct='%1.1f%%', startangle=140)
    plt.title('Distribution of Heart Condition')
    plt.show()
```

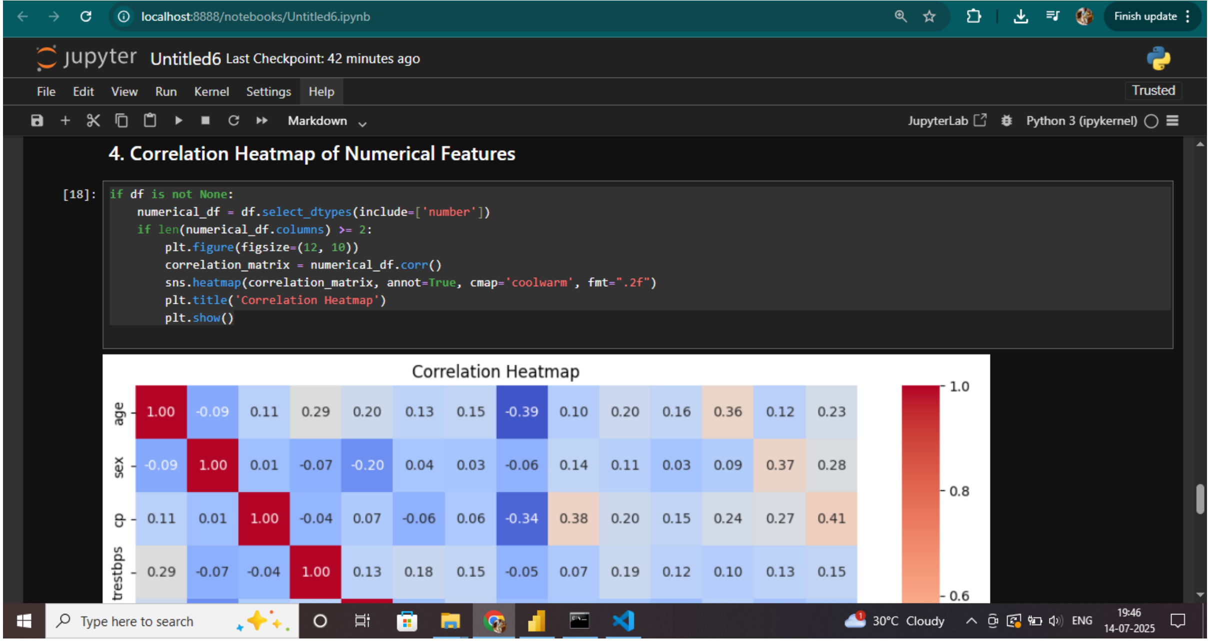
Distribution of Heart Condition
1


localhost:8888/notebooks/Untitled6.ipynb
Finish update
jupyter Untitled6 Last Checkpoint: 42 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Markdown
JupyterLab
Python 3 (ipykernel)

2. Age vs. Maximum Heart Rate by Condition

[16]:
```
if df is not None and 'age' in df.columns and 'thalach' in df.columns:
    plt.figure(figsize=(10, 6))
    sns.scatterplot(data=df, x='age', y='thalach', hue='condition')
    plt.title('Scatter Plot of Age vs. Maximum Heart Rate')
    plt.xlabel('Age')
    plt.ylabel('Maximum Heart Rate Achieved')
    plt.show()
```

Scatter Plot of Age vs. Maximum Heart Rate
condition
0
1
200
180
160
140
rt Rate Achieved

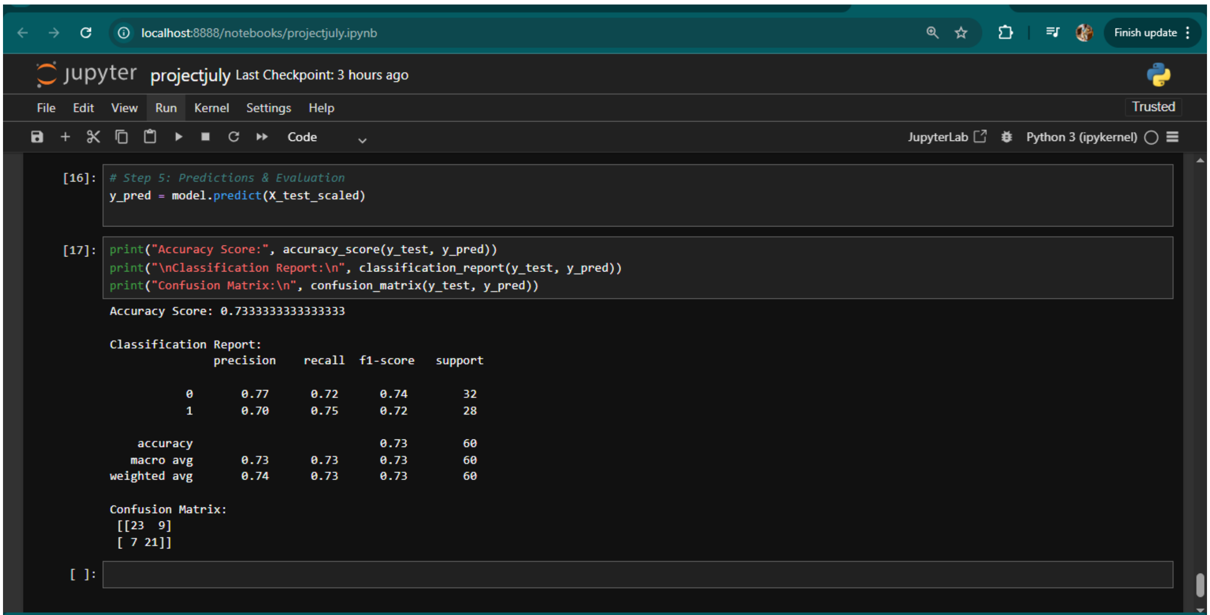

localhost:8888/notebooks/Untitled6.ipynb
Finish update
jupyter Untitled6 Last Checkpoint: 42 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Markdown
JupyterLab
Python 3 (ipykernel)

4. Correlation Heatmap of Numerical Features

[18]:
```
if df is not None:
    numerical_df = df.select_dtypes(include=['number'])
    if len(numerical_df.columns) >= 2:
        plt.figure(figsize=(12, 10))
        correlation_matrix = numerical_df.corr()
        sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
        plt.title('Correlation Heatmap')
        plt.show()
```

Correlation Heatmap
age 1.00 -0.09 0.11 0.29 0.20 0.13 0.15 -0.39 0.10 0.20 0.16 0.36 0.12 0.23
sex -0.09 1.00 0.01 -0.07 -0.20 0.04 0.03 -0.06 0.14 0.11 0.03 0.09 0.37 0.28
cp 0.11 0.01 1.00 -0.04 0.07 -0.06 0.06 -0.34 0.38 0.20 0.15 0.24 0.27 0.41
trestbps 0.29 -0.07 -0.04 1.00 0.13 0.18 0.15 -0.05 0.07 0.19 0.12 0.10 0.13 0.15
1.0
0.8
0.6

Type here to search
30°C Cloudy
ENG
19:46
14-07-2025

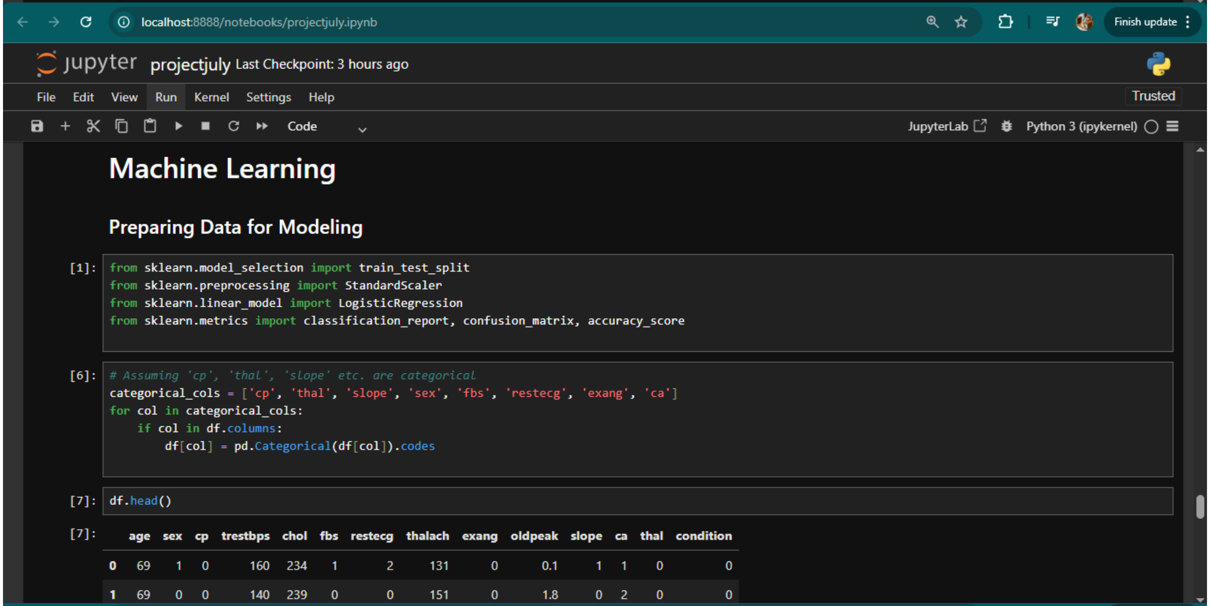

localhost:8888/notebooks/projectjuly.ipynb
Finish update
jupyter projectjuly Last Checkpoint: 3 hours ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)

```
[16]: # Step 5: Predictions & Evaluation
      y_pred = model.predict(X_test_scaled)
```

```
[17]: print("Accuracy Score:", accuracy_score(y_test, y_pred))
      print("\nClassification Report:\n", classification_report(y_test, y_pred))
      print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

```
Accuracy Score: 0.7333333333333333

Classification Report:
               precision    recall  f1-score   support

           0       0.77      0.72      0.74        32
           1       0.70      0.75      0.72        28

    accuracy                           0.73        60
   macro avg       0.73      0.73      0.73        60
weighted avg       0.74      0.73      0.73        60

Confusion Matrix:
 [[23  9]
 [ 7 21]]
```

```
[ ]:
```


localhost:8888/notebooks/projectjuly.ipynb
Finish update
jupyter projectjuly Last Checkpoint: 3 hours ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)

Machine Learning

Preparing Data for Modeling

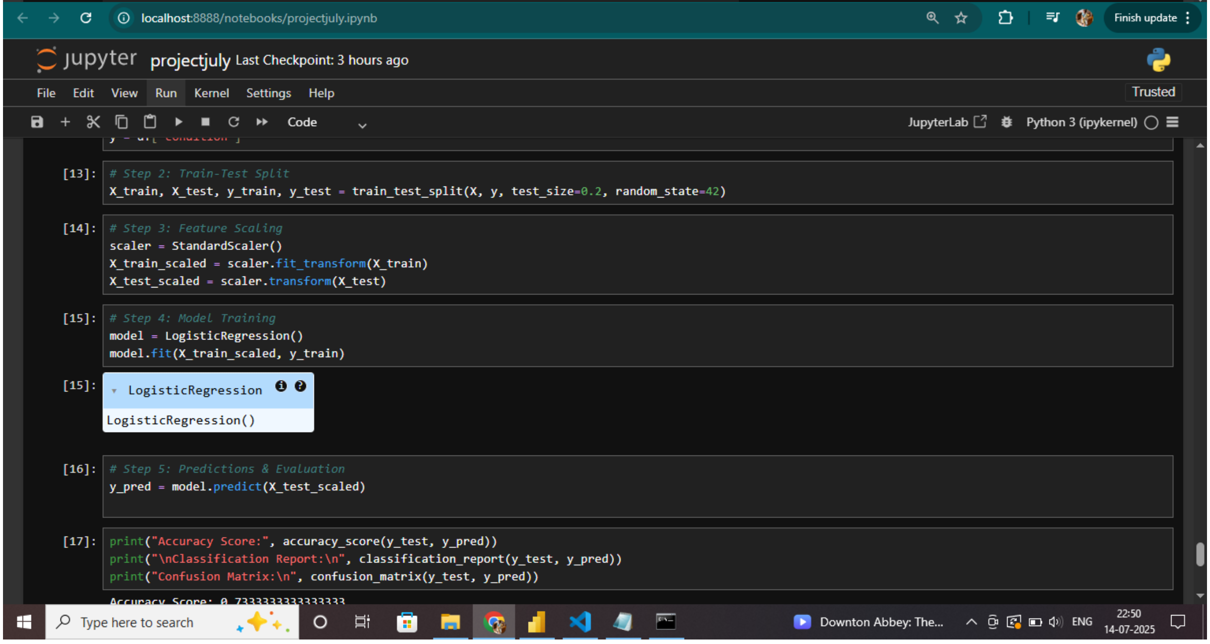
```
[1]: from sklearn.model_selection import train_test_split
     from sklearn.preprocessing import StandardScaler
     from sklearn.linear_model import LogisticRegression
     from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
```

```
[6]: # Assuming 'cp', 'thal', 'slope' etc. are categorical
     categorical_cols = ['cp', 'thal', 'slope', 'sex', 'fbs', 'restecg', 'exang', 'ca']
     for col in categorical_cols:
         if col in df.columns:
             df[col] = pd.Categorical(df[col]).codes
```

```
[7]: df.head()
```

[7]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	condition
0	69	1	0	160	234	1	2	131	0	0.1	1	1	0	0
1	69	0	0	140	239	0	0	151	0	1.8	0	2	0	0


localhost:8888/notebooks/projectjuly.ipynb
Finish update
jupyter projectjuly Last Checkpoint: 3 hours ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Type here to search
Downton Abbey: The...
ENG
22:50
14-07-2025

```
y = df['condition']
```

```
[13]: # Step 2: Train-Test Split
      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
[14]: # Step 3: Feature Scaling
      scaler = StandardScaler()
      X_train_scaled = scaler.fit_transform(X_train)
      X_test_scaled = scaler.transform(X_test)
```

```
[15]: # Step 4: Model Training
      model = LogisticRegression()
      model.fit(X_train_scaled, y_train)
```

```
[15]: ▾ LogisticRegression
      LogisticRegression()
```

```
[16]: # Step 5: Predictions & Evaluation
      y_pred = model.predict(X_test_scaled)
```

```
[17]: print("Accuracy Score:", accuracy_score(y_test, y_pred))
      print("\nClassification Report:\n", classification_report(y_test, y_pred))
      print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
      Accuracy Score: 0.7333333333333333
```

CHAPTER 5: RESULTS AND DISCUSSION

Output / Report

The analysis of the heart disease dataset revealed several key insights that are critical in understanding the factors contributing to heart disease. Among the many features studied, chest pain type, maximum heart rate achieved (thalach), and ST depression induced by exercise (oldpeak) stood out as having the strongest correlations with the presence of heart disease.

- Patients with atypical or non-anginal chest pain types showed a significantly higher likelihood of heart disease compared to those with typical angina.
- Lower maximum heart rate was commonly observed in patients with heart disease, suggesting reduced cardiovascular performance.
- Higher values of ST depression (oldpeak) were also directly associated with an increased risk of heart disease.

These findings were supported both visually (through count plots, box plots, and heatmaps) and statistically (via correlation analysis). The results were compiled into a Power BI dashboard, where end users can filter and interact with the data to explore heart disease risk by gender, age group, and more.

Challenges Faced

Some data was imbalanced across classes. Handling outliers and interpreting overlapping features required domain understanding. Throughout the course of the project, the team encountered several challenges that required critical thinking and collaborative problem-solving:

- Class Imbalance: The target variable (presence of heart disease) was not evenly distributed, which initially impacted the visual analysis and interpretation. We had to ensure that results weren't biased due to this imbalance.
- Handling Outliers: Some numerical features like cholesterol and resting blood pressure showed unusually high or low values. Deciding whether to remove or retain these outliers required domain knowledge and caution to avoid misrepresentation.
- Feature Interpretation: Understanding medical terms like ST depression, chest pain types, and their real-world significance involved background research in the healthcare domain, which was essential for making meaningful conclusions.
- Power BI Learning Curve: As most members were first-time Power BI users, learning to design a clean, interactive dashboard took additional time and multiple iterations.

Despite these challenges, the team adapted well and was able to produce high-quality outcomes through consistent efforts and teamwork.

Learnings

This project was an enriching experience that led to both technical and domain-specific learning:

- Hands-on Experience with EDA: The team gained a deep understanding of how to perform and interpret Exploratory Data Analysis using Python libraries like Pandas, Matplotlib, and Seaborn.
- Dashboard Design and Data Storytelling: We learned to convert complex datasets into simple, meaningful visuals using Power BI. This helped develop our ability to tell a compelling story with data.
- Domain Knowledge in Healthcare Analytics: Through the interpretation of health features and their impact on heart disease, we gained valuable insights into the intersection of data science and healthcare.
- Team Collaboration and Report Compilation: Finally, the experience of working as a team and compiling a formal academic report improved our collaboration, documentation, and presentation skills.

CHAPTER 6: CONCLUSION

Summary

This project provided a comprehensive exploration into the application of data analytics in the healthcare sector, specifically focusing on heart disease prediction. By leveraging Exploratory Data Analysis (EDA) techniques, the team was able to examine a real-world dataset containing various patient health indicators and uncover meaningful patterns related to cardiovascular risk.

The process began with data cleaning and preprocessing, followed by in-depth analysis using statistical summaries and visualizations. Features such as chest pain type, maximum heart rate, and ST depression were found to be the most indicative of heart disease. These insights were not only derived through code-based analysis in Python but also translated into an interactive Power BI dashboard, making them accessible to non-technical stakeholders as well.

This project served as a strong learning foundation in areas like data wrangling, visualization, and domain-specific analysis. More importantly, it demonstrated how healthcare data, when properly analyzed and visualized, can guide early intervention and improved decision-making.

The work done in this project also opens the door to future enhancements, such as developing predictive machine learning models using the same dataset. With further optimization and model training, this project could evolve into a more automated heart disease risk classification tool.

Overall, the project not only fulfilled its objective of understanding heart disease patterns but also helped bridge the gap between raw data and actionable healthcare insights.