KUMARAGURU COLLEGE OF TECHNOLOGY

DEPARTMENT OF ARTIFICAL INTELGINCE AND DATA SCIENCE

24AD1204 – DATA SCIENCE AND VISULIZATION

NAME	ROLLNO
GOKULNAATH M	24BAD026
JAYARAKSHA REGURAJ	24BAD044
KAMALESH N	24BAD054
NISHANTH P	24BAD405

Submitted to
Mrs. E Shriaarthy

Week No: 1 **Team Formation & Data Hunting**

Objectives:

Selection of dataset. Setting up GitHub. Introduction to Pandas/NumPy.

Team Composition:

NAME	ROLES
GOKULNAATH M	DATA ENGINEER
JAYARAKSHA REGURAJ	STORYTELLER
KAMALESH N	DATA ANALYST
NISHANTH P	DATA VISUALIZATION

Work done:

1. Domain Selection and Problem Statement Abstract

Team Contribution:

1. Dataset Selection & Evaluation

Gokulnaath M – 24BAD028

Dataset 1

Heart Attack Prediction Dataset

https://www.kaggle.com/datasets/iamsouravbanerjee/heart-attack-prediction-dataset

Characteristics

- Low / limited attributes
- Low object (record) count
- Semi-structured data

Pros

- Uncleaned dataset – ideal for demonstrating data preprocessing, handling missing values, and noise reduction
- Simple structure, easy to understand feature relationships
- Good for baseline model development

Cons

- Limited number of attributes restricts deep risk factor analysis
- Small dataset size reduces model generalization
- Not suitable alone for dimensionality reduction showcase

Jayaraksha Reguraj – 24BAD044

Dataset 2

Heart Disease Dataset

https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset

Characteristics

- Already cleaned and pre-processed
- Low / limited object count

Pros

- Clean and ready for quick model training
- Good for algorithm comparison
- Minimal preprocessing effort required

Cons

- Already cleaned – not ideal for project requirements that demand full preprocessing pipeline
- Less opportunity to demonstrate data cleaning skills
- Limited data volume

Kamalesh N – 24BAD054

Dataset 3 (Finalized Dataset)

Heart Disease Prediction using Logistic Regression

https://www.kaggle.com/datasets/dileep070/heart-disease-prediction-using-logistic-regression

Characteristics

- Good number of attributes
- Dataset is not pre-processed

Pros

- Uncleaned dataset – perfectly aligned with project requirement to perform:
    - Data cleaning
    - Feature scaling
    - Dimensionality reduction (PCA)
- Balanced attribute set supports risk score generation
- Suitable for end-to-end ML pipeline

Cons

- Requires significant preprocessing effort
- Noise and missing values may affect initial model performance
- Needs careful feature engineering

Nishanth P – 24BAD405

Dataset 4
Framingham Heart Study Dataset
https://www.kaggle.com/datasets/amanajmera1/framingham-heart-study-dataset
Characteristics

- Long-term cardiovascular study data
- Moderate to high attribute count
- Includes demographic, behavioural, and medical risk factors

Pros

- Raw / uncleaned dataset – excellent for showcasing:
  Missing value handling
  Outlier detection
  Feature transformation
- Rich clinical depth enables strong risk factor interpretation
- Ideal for dimensionality reduction & risk score modeling

Cons

- Requires extensive preprocessing
- Some attributes may need domain understanding
- Higher complexity compared to smaller datasets

Summary Comparison Table

Dataset	Pre-processed	Attribute Count	Object Count	Suitability
Dataset 1	No	Low	Low	Basic preprocessing demo
Dataset 2	Yes	Low	Low	Model testing only
Dataset 3 (Final)	No	High	Medium	Best fit for project
Dataset 4	No	High	Medium	Advanced analysis

2. Setting up GitHub
All *
Procedure

- Create GitHub Repositories with name "**24ADI204_DSV_Team14**"
- Add Description of
  "Heartbeats & Habits" combats the silent threat of cardiovascular disease by applying rigorous cleaning and dimensionality reduction to noisy clinical data. This approach transforms complex biomarker and lifestyle interactions into a unified "Risk Score," visualizing a clear path to improved cardiac health
- Add README.md

3. Introduction to Pandas/NumPy
All *
Procedure

- Research and learn the basic concept of NumPy
- Research and learn the basic concept of Pandas

Resources and References

- **Dataset**
  - https://www.kaggle.com/datasets/iamsouravbanerjee/heart-attack-prediction-dataset
  - https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset
  - https://www.kaggle.com/datasets/dileep070/heart-disease-prediction-using-logistic-regression
  - https://www.kaggle.com/datasets/amanajmera1/framingham-heart-study-dataset

- **GitHub**
  - https://github.com/Gokulnaath-gif/24ADI204_DSV_Team14

- **Learns**
  - https://numpy.org/doc/stable/user/index.html#user
  - https://www.w3schools.com/python/numpy/numpy_intro.asp
  - https://pandas.pydata.org/docs/user_guide/index.html
  - https://www.w3schools.com/python/pandas/default.asp

Week No: 2 **Know Your Data**

Objectives:

Loading data, checking types, inspecting the structure.

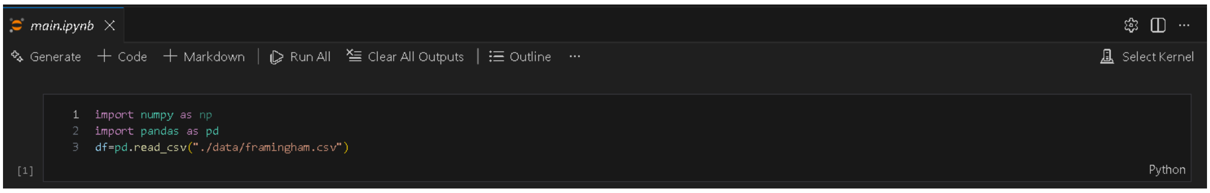
Work done:

1. Load Data and Analyses Dataset

Gokulnaath M – 24BAD028, Nishanth P – 24BAD405

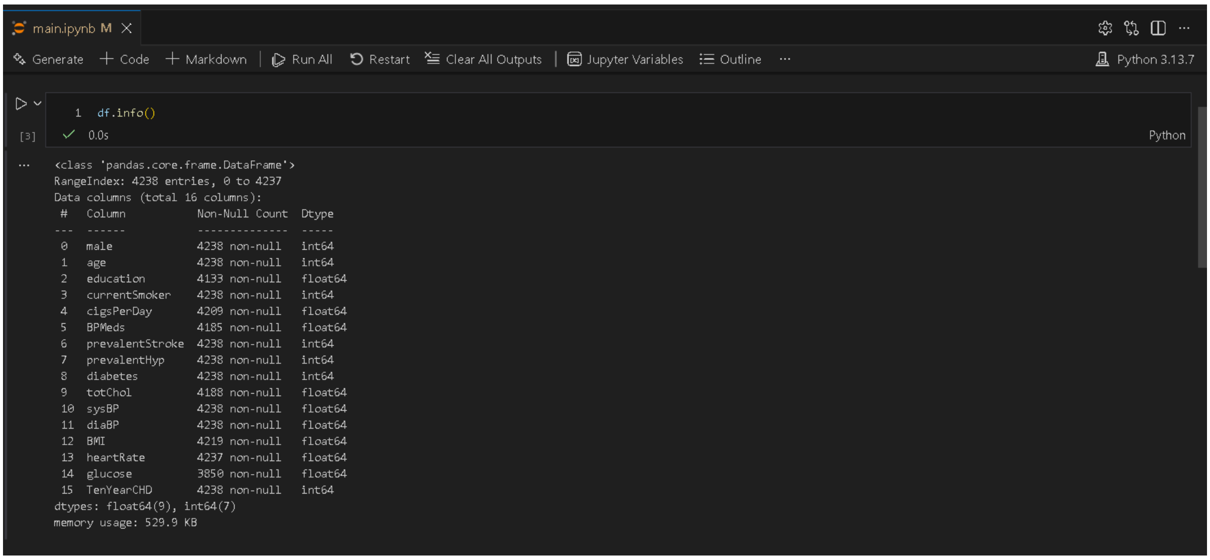
Procedure

1. Load the Dataset



```python
import numpy as np
import pandas as pd
df=pd.read_csv("./data/framingham.csv")
```

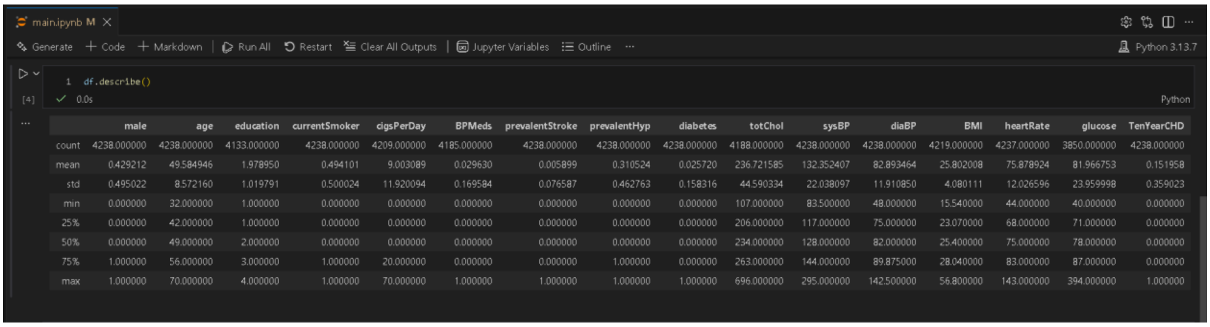
2. Data Structure



```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4238 entries, 0 to 4237
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   male             4238 non-null   int64
 1   age              4238 non-null   int64
 2   education        4133 non-null   float64
 3   currentSmoker    4238 non-null   int64
 4   cigsPerDay       4209 non-null   float64
 5   BPMeds           4185 non-null   float64
 6   prevalentStroke  4238 non-null   int64
 7   prevalentHyp     4238 non-null   int64
 8   diabetes         4238 non-null   int64
 9   totChol          4188 non-null   float64
 10  sysBP            4238 non-null   float64
 11  diaBP            4238 non-null   float64
 12  BMI              4219 non-null   float64
 13  heartRate        4237 non-null   float64
 14  glucose          3850 non-null   float64
 15  TenYearCHD       4238 non-null   int64
dtypes: float64(9), int64(7)
memory usage: 529.9 KB
```

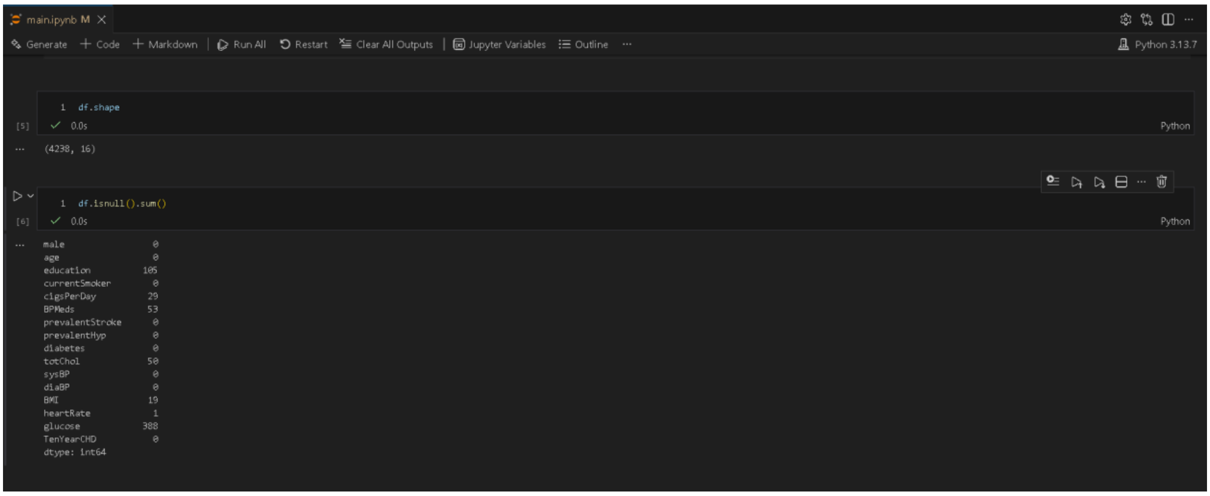
3. Data Statistical Information



```
df.describe()
```

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRate	glucose	TenYearCHD
count	4238.000000	4238.000000	4133.000000	4238.000000	4209.000000	4185.000000	4238.000000	4238.000000	4238.000000	4188.000000	4238.000000	4238.000000	4219.000000	4237.000000	3850.000000	4238.000000
mean	0.429212	49.584946	1.978950	0.494101	9.003089	0.029630	0.005899	0.310524	0.025720	236.721585	132.352407	82.893464	25.802008	75.878924	81.966753	0.151958
std	0.495022	8.572160	1.019791	0.500024	11.920094	0.169584	0.076587	0.462763	0.158316	44.590334	22.038097	11.910850	4.080111	12.026596	23.959998	0.359023
min	0.000000	32.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	107.000000	83.500000	48.000000	15.540000	44.000000	40.000000	0.000000
25%	0.000000	42.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	206.000000	117.000000	75.000000	23.070000	68.000000	71.000000	0.000000
50%	0.000000	49.000000	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	234.000000	128.000000	82.000000	25.400000	75.000000	78.000000	0.000000
75%	1.000000	56.000000	3.000000	1.000000	20.000000	0.000000	0.000000	1.000000	0.000000	263.000000	144.000000	89.875000	28.040000	83.000000	87.000000	0.000000
max	1.000000	70.000000	4.000000	1.000000	70.000000	1.000000	1.000000	1.000000	1.000000	696.000000	295.000000	142.500000	56.800000	143.000000	394.000000	1.000000

4. Data Structure and Inspecting Null Values



```
df.shape
```

```
(4238, 16)
```

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

```
male                 0
age                  0
education          105
currentSmoker        0
cigsPerDay          29
BPMeds              53
prevalentStroke      0
prevalentHyp         0
diabetes             0
totChol             50
sysBP                0
diaBP                0
BMI                 19
heartRate            1
glucose            388
TenYearCHD           0
dtype: int64
```

2. "First Impression" report.

Jayaraksha Reguraj – 24BAD044, Kamalesh N – 24BAD054

Report

1. Dataset Overview

The dataset used for this analysis is the Framingham Heart Study dataset, which contains medical and lifestyle information of patients. The primary objective of the dataset is to predict the 10-year risk of coronary heart disease (CHD).

- Number of records: 4,240
- Number of features: 16
- Target variable: TenYearCHD (binary: 0 = No CHD, 1 = CHD)

2. Data Structure

- The dataset consists of numerical features (both integer and float types).
- Variables include:

  Demographic features: age, sex

  Behavioural features: smoking status, cigarettes per day

  Medical history: blood pressure medication, diabetes, stroke history

Clinical measurements: cholesterol, systolic & diastolic BP, BMI, heart rate, glucose

3. Data Types

- Most features are of type int64 or float64.
- The target variable TenYearCHD is an integer categorical variable representing a binary outcome.
- No categorical string variables are present, reducing the need for encoding at this stage.

4. Missing Values

- Several columns contain missing values, including:
  - education
  - cigsPerDay
  - BPMeds
  - totChol
  - BMI
  - heartRate
  - glucose

5. Initial Observations

- The dataset appears realistic and noisy, which is expected in medical data.
- Feature scales vary significantly (e.g., age vs. cholesterol vs. glucose), suggesting the need for feature scaling before modeling.
- The target variable is imbalanced, with fewer positive CHD cases compared to negative ones, which may affect model performance.

6. Data Quality Assessment

- No duplicate rows were observed.
- No obvious data corruption or invalid values were detected during initial inspection.
- Presence of missing values is the main data quality concern.

7. Overall, First Impression

The dataset is:

- Well-structured
- Relevant for classification tasks
- Contains missing values
- Requires preprocessing and scaling
- Potential class imbalance in the target variable

Resources and References

- **Dataset**
  - https://www.kaggle.com/datasets/dileep070/heart-disease-prediction-using-logistic-regression

- **GitHub**
  - https://github.com/Gokulnaath-gif/24ADI204_DSV_Team14

Week No: 3 **The Cleaning Sprint**

Objectives:

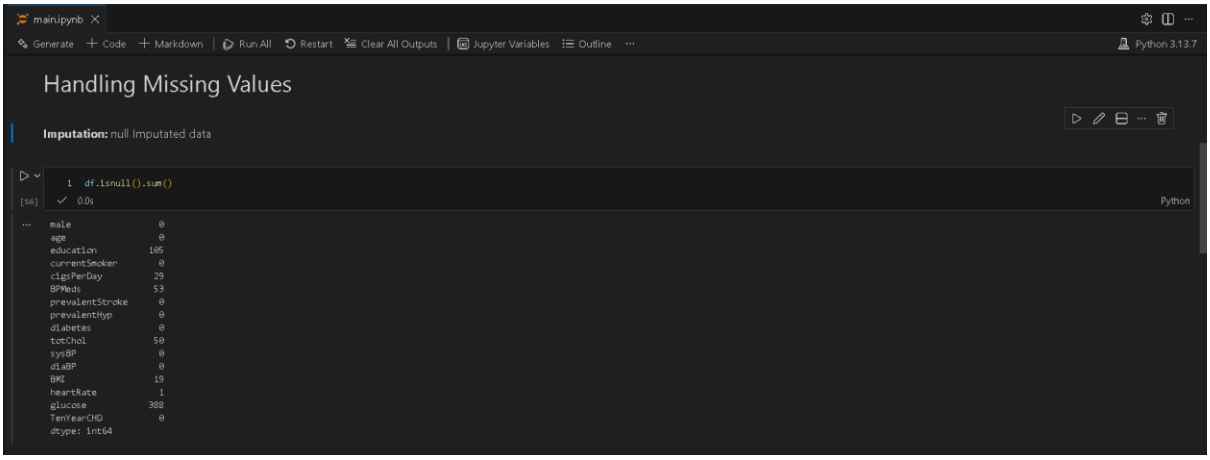
Handling Missing Values (Imputation strategies) and Outlier Detection (Boxplots/Z-score).

Work done:

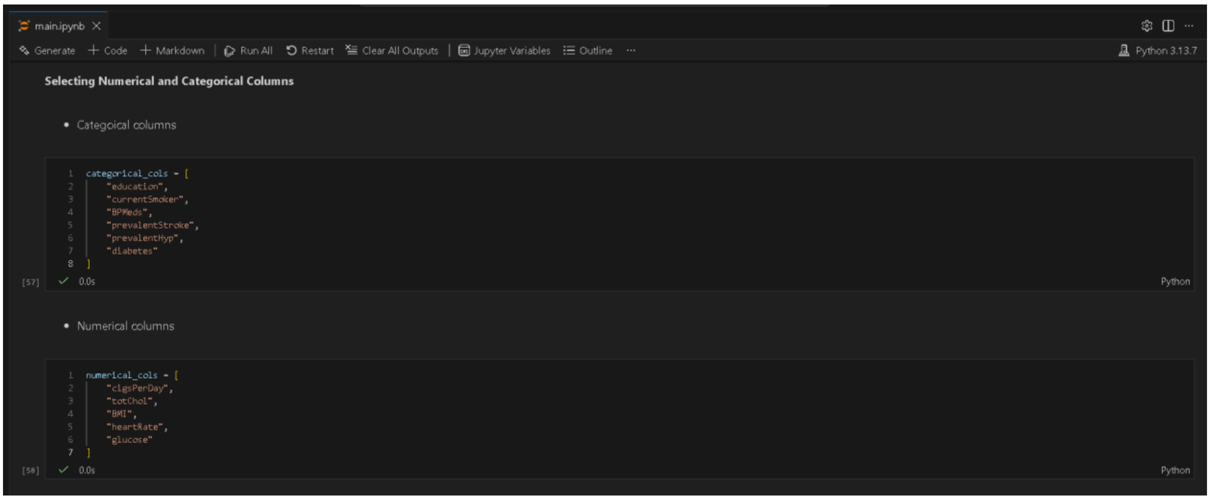
1. Handling Null Values

Gokulnaath M – 24BAD028, Nishanth P – 24BAD405

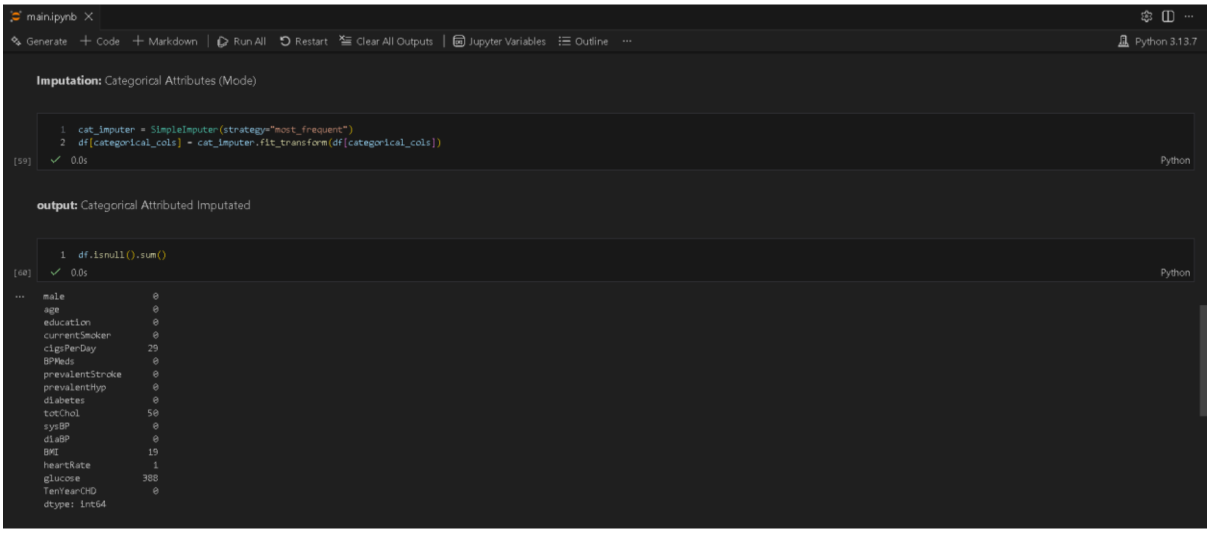
1. Attributes Contains Null Values



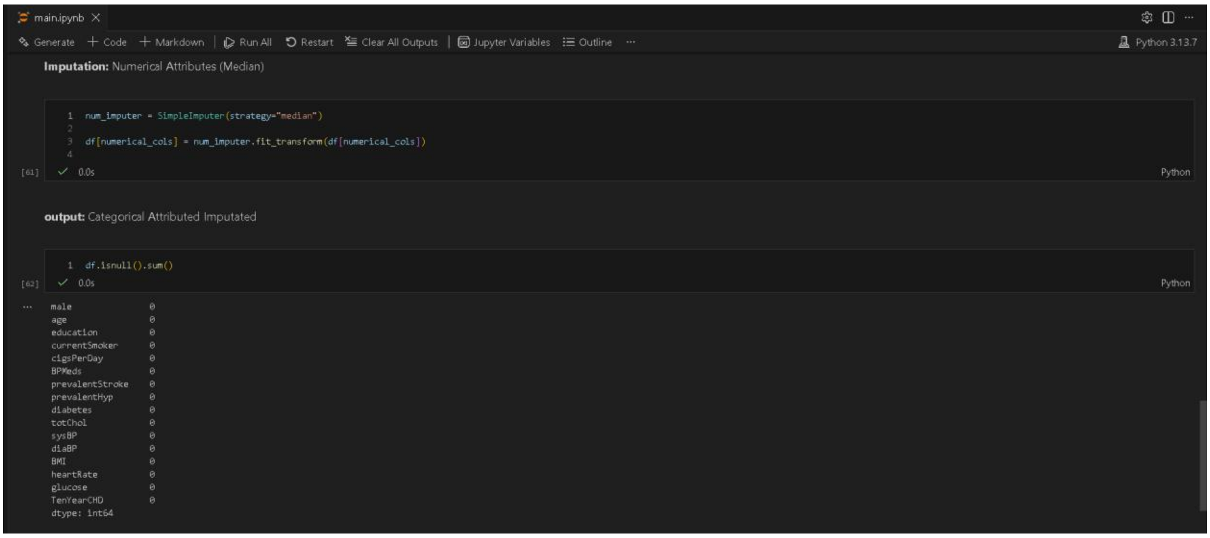
2. Selecting Categorical and Numerical Attributes



3. Imputation of Categorical Attributes (Mode)



4. Imputation of Numerical Attributes (Median)



2. Handling Outliers values

Jayaraksha Reguraj – 24BAD044, Kamalesh N – 24BAD054

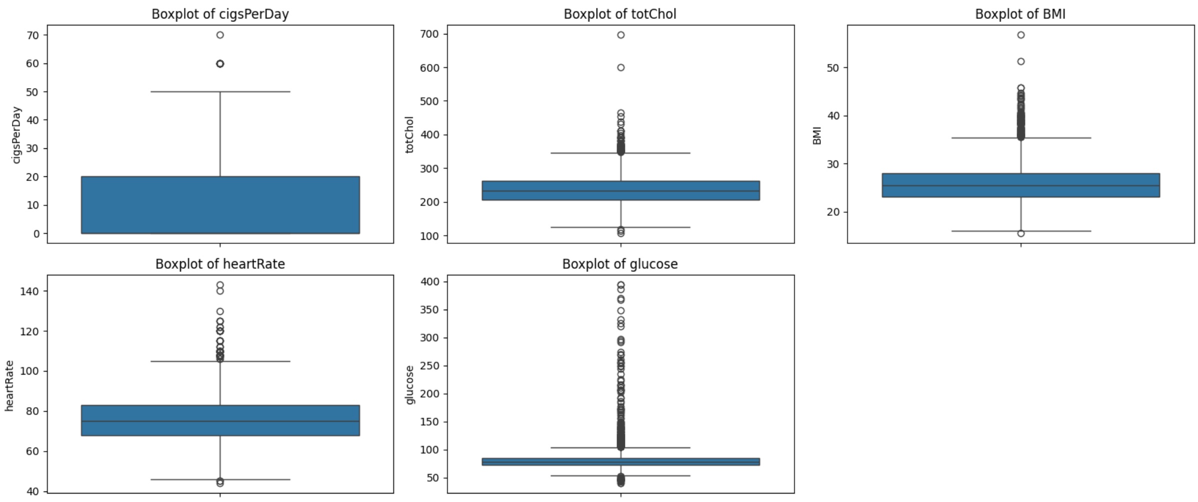
1. Outlier Detections using Boxplot

Outlier Dectection: Numerical attributes

- Using Boxplot

```python
plt.figure(figsize=(16, 10))

for i, col in enumerate(numerical_cols, 1):
    plt.subplot(3, 3, i)
    sns.boxplot(y=df[col])
    plt.title(f"Boxplot of {col}")

plt.tight_layout()
plt.show()

```

[63] 0.9s Python



2. Outlier Detections using Z-Score

main.ipynb

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· Python 3.13.7

- Using Z-Score

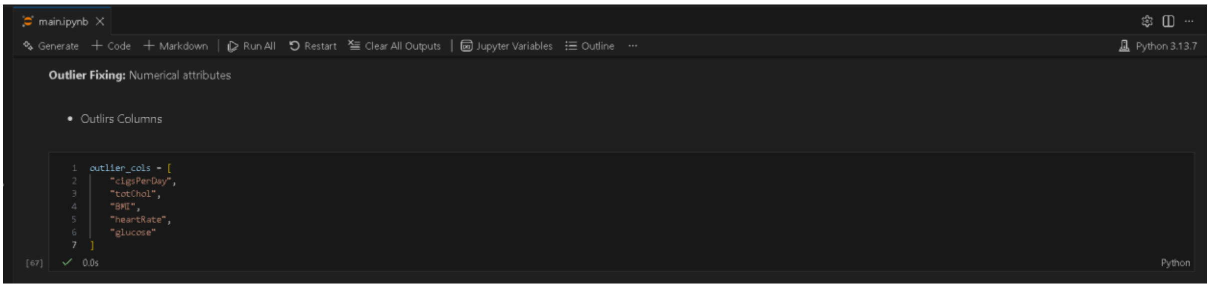
```python
z_scores = df[numerical_cols].apply(zscore)

outliers = (np.abs(z_scores) > 3)

outlier_count = outliers.sum()
outlier_count

```

[64] 0.0s Python

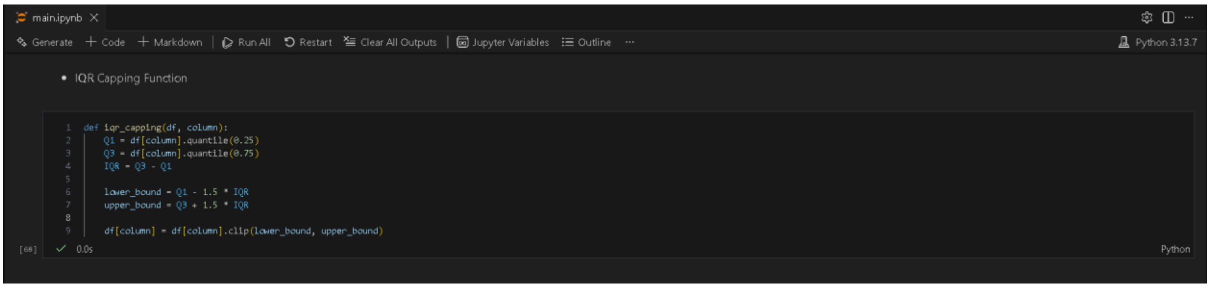
```
cigsPerDay    21
totChol       25
BMI           60
heartRate     23
glucose       50
dtype: int64
```

3. Outlier Fixing

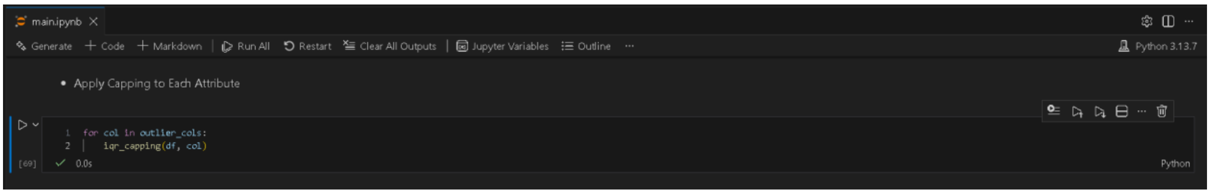
- Select Outlier Columns



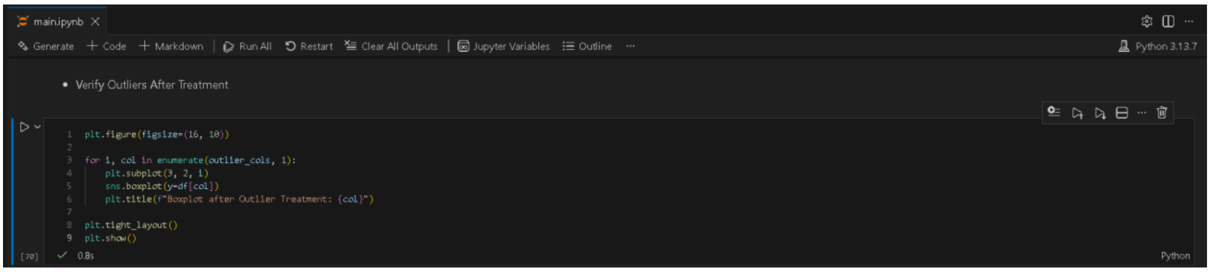
- IQR Capping Function



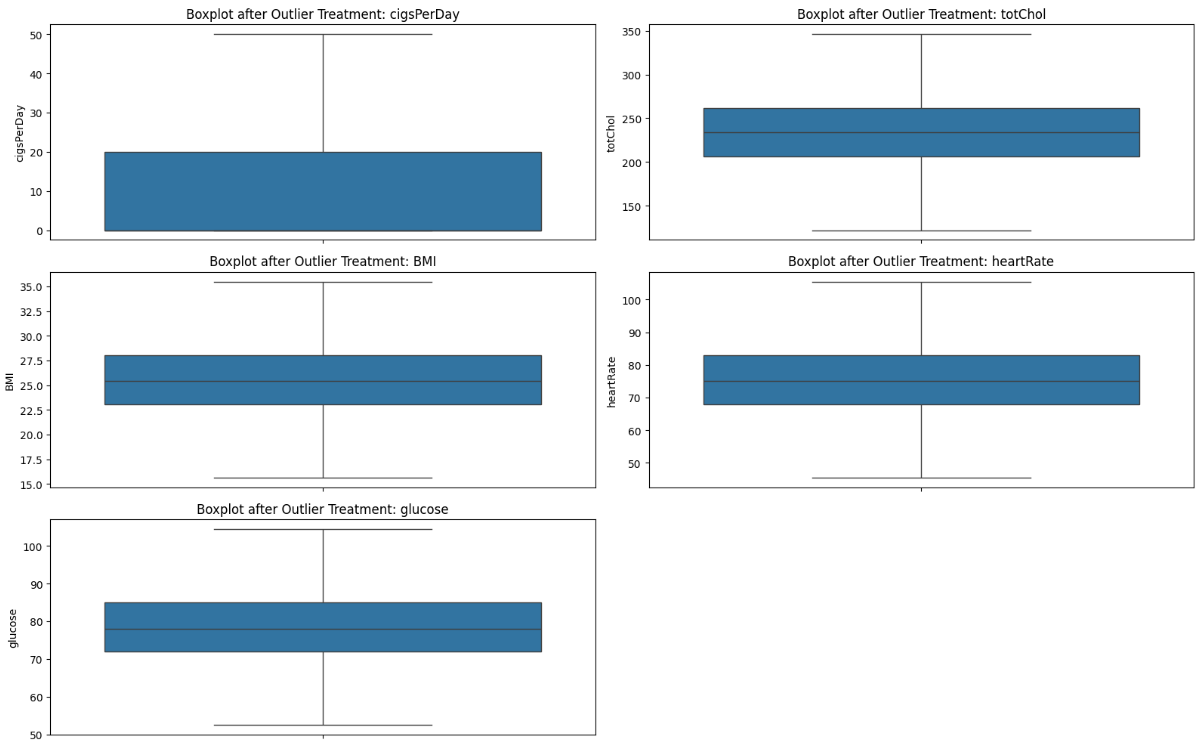
- Apply Capping to Each Attribute



- Verify Outliers After Treatment




```python
plt.figure(figsize=(16, 10))

for i, col in enumerate(outlier_cols, 1):
    plt.subplot(3, 2, i)
    sns.boxplot(y=df[col])
    plt.title(f"Boxplot after Outlier Treatment: {col}")

plt.tight_layout()
plt.show()
```



Resources and References

- **Dataset**
  - https://www.kaggle.com/datasets/dileep070/heart-disease-prediction-using-logistic-regression

- **GitHub**
  - https://github.com/Gokulnaath-gif/24ADI204_DSV_Team14

- **Resources**
  - https://www.geeksforgeeks.org/data-analysis/working-with-missing-data-in-pandas/
  - https://medium.com/@punya8147_26846/dataframes-handling-missing-values-in-pandas-11f7702afaf7
  - https://www.geeksforgeeks.org/data-science/detect-and-remove-the-outliers-using-python/
  - https://www.geeksforgeeks.org/pandas/handling-outliers-with-pandas/
  - https://llego.dev/posts/outlier-detection-handling-python-guide/
  - https://x.ai/grok
  - https://openai.com/chatgpt

Week No: 4 **EDA Deep Dive**

Objectives:

Variant analysis and Visualizing distributions.

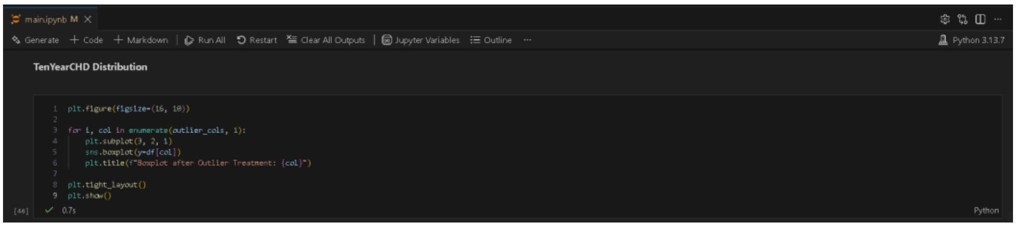
Work done:

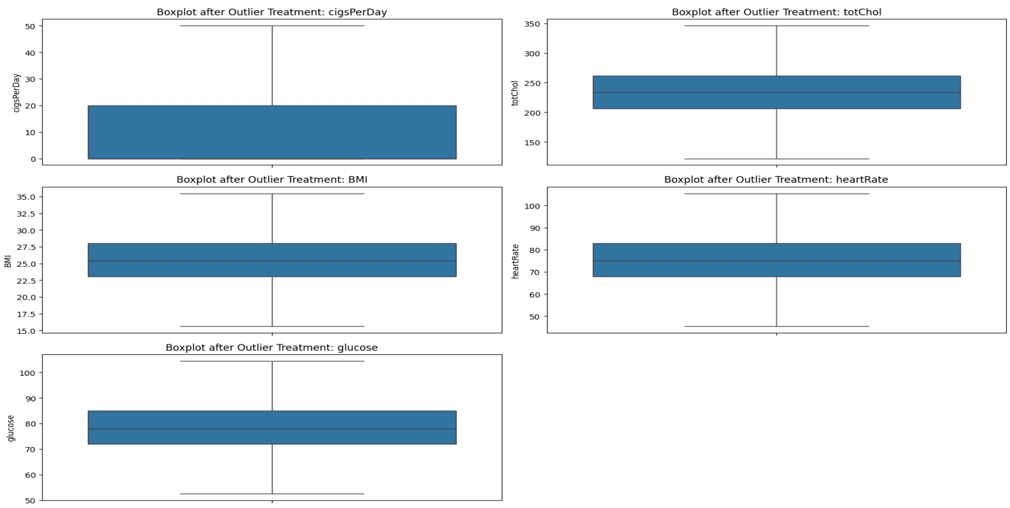
1. Variant Analysis

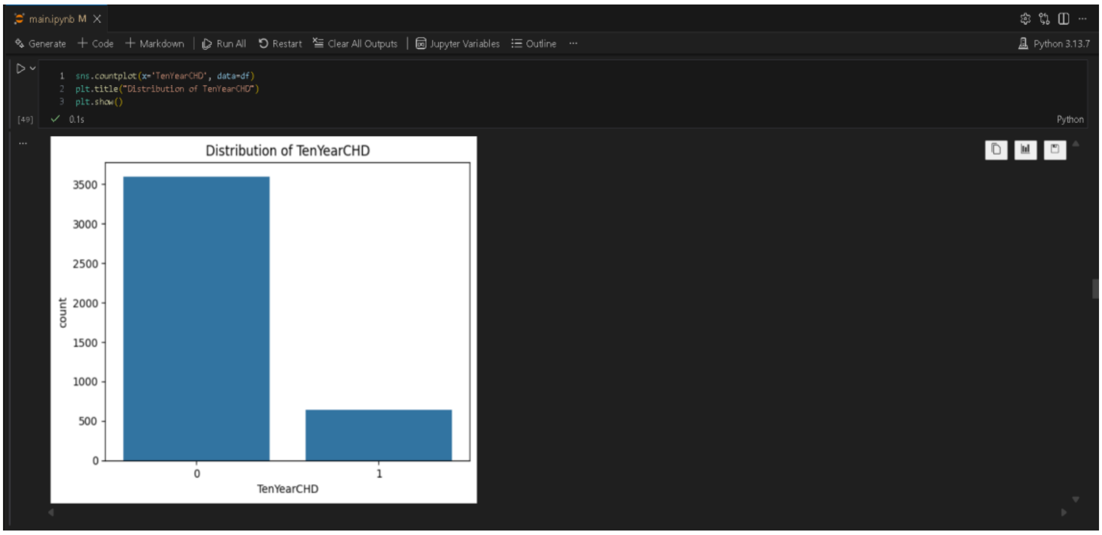
Kamalesh N 24BAD054 - Nishanth P 24BAD405

1. Univariate

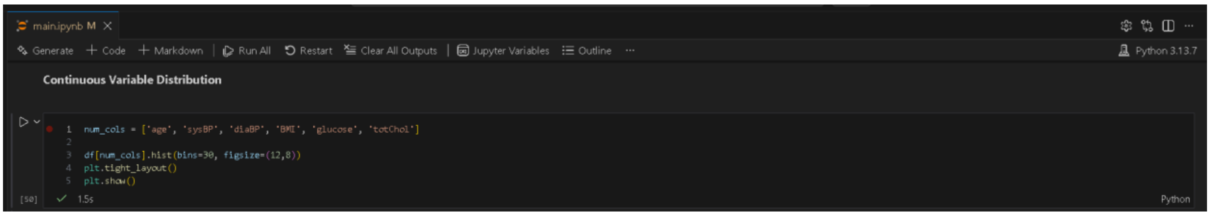
1. TenYearCHD Distribution

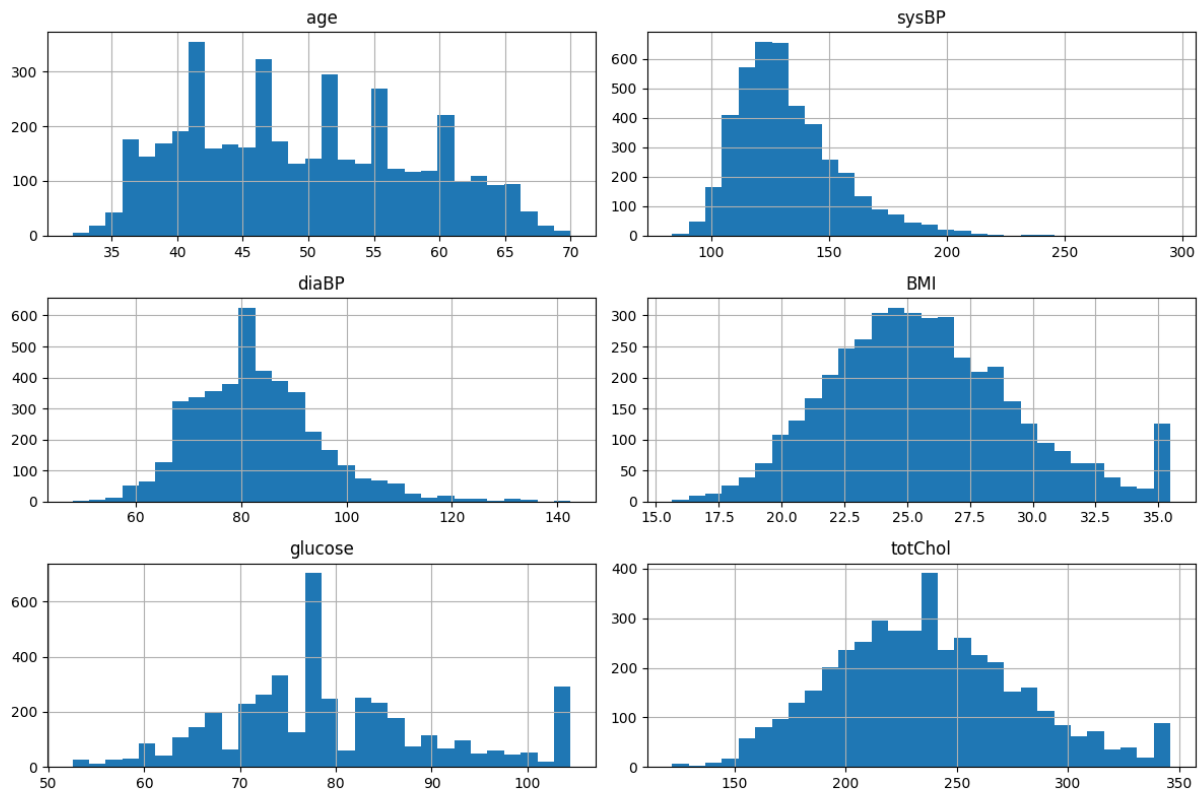






2. Continuous Variable Distribution





Why Histogram?

- Shows distribution shape (Normal / Skewed)
- Helps identify skewness

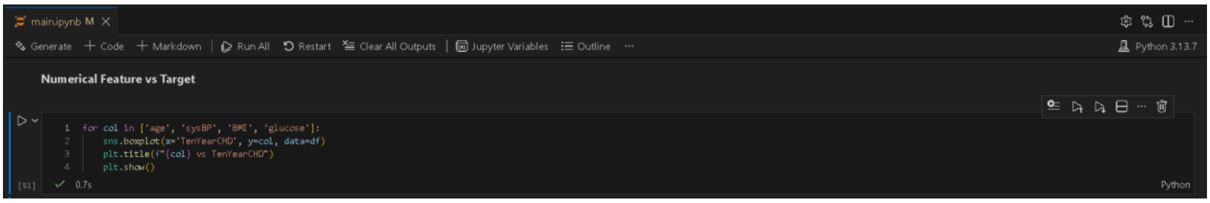
- Helps check extreme values

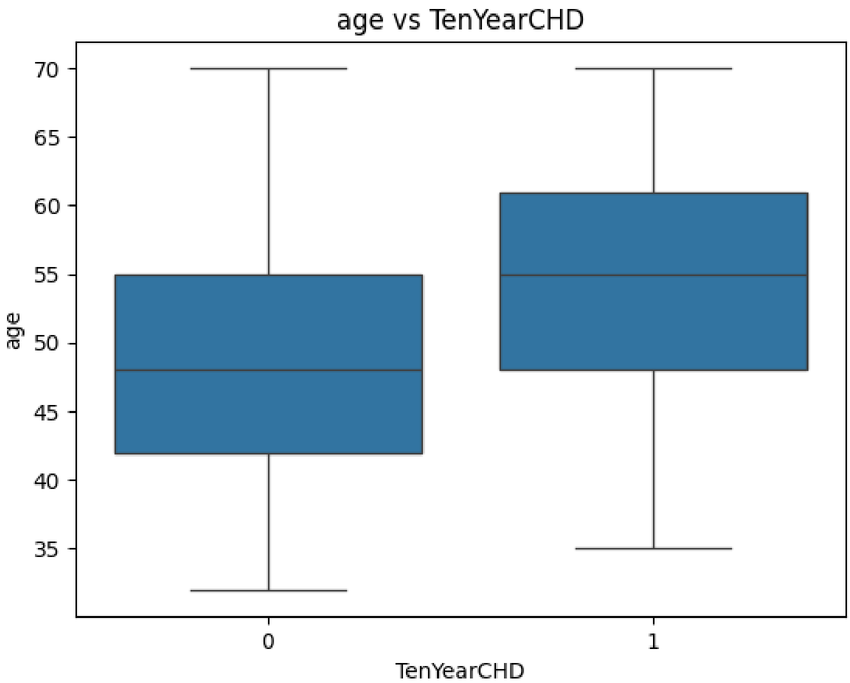
Univariate Observations

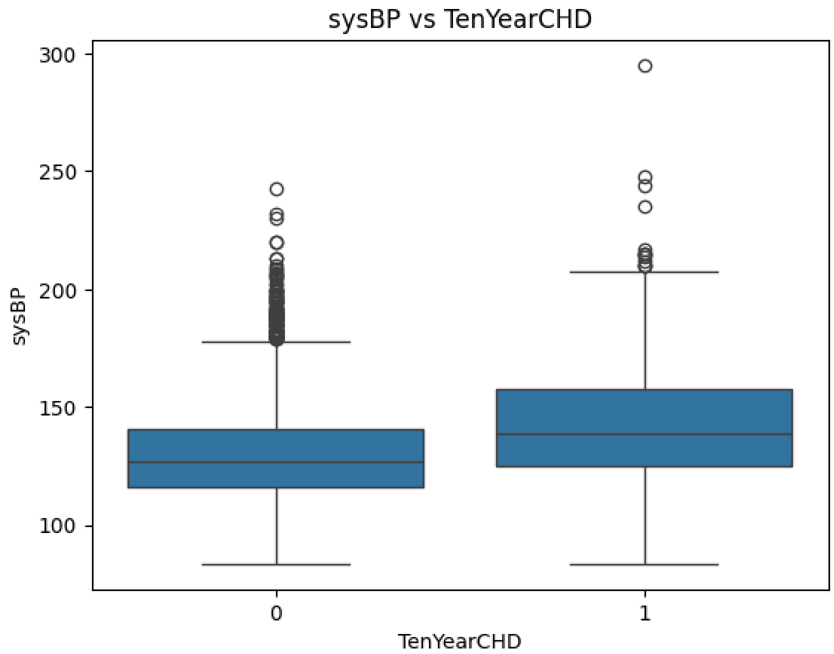
- age, BMI, glucose show right-skewed distributions
- sysBP and diaBP are approximately normal
- Outliers present in glucose and cholesterol

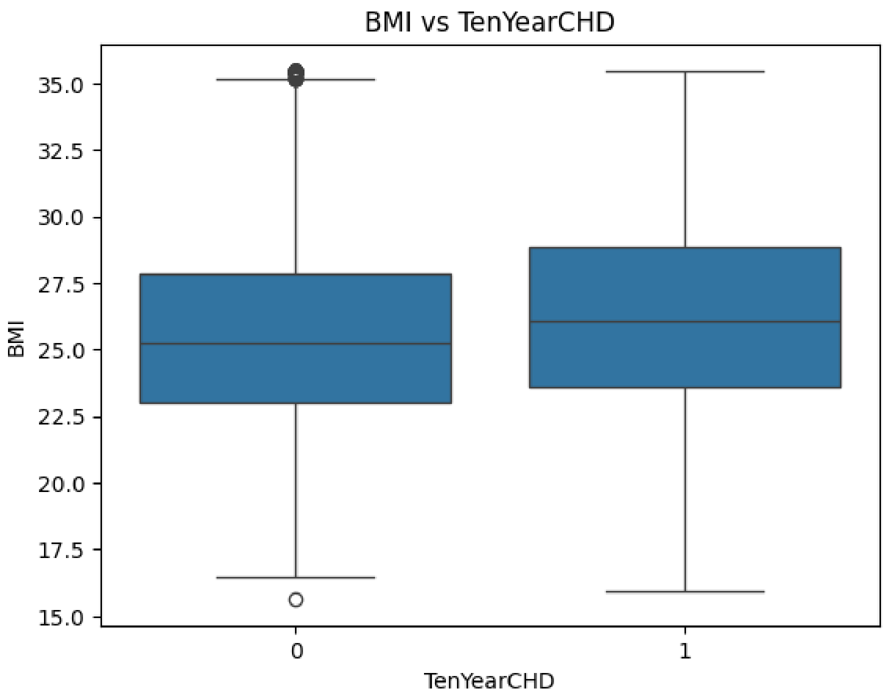
2. Bivariate

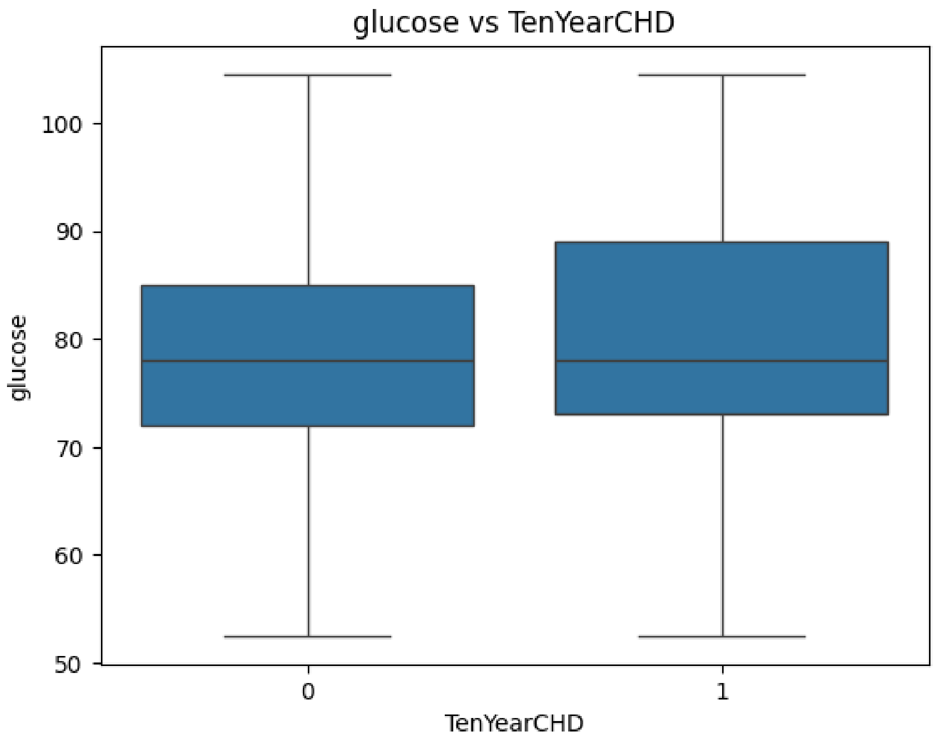
1. Numerical Feature vs Target






sysBP vs TenYearCHD
300
250
200
150
100
sysBP
0
1
TenYearCHD


BMI vs TenYearCHD
35.0
32.5
30.0
27.5
25.0
22.5
20.0
17.5
15.0
BMI
0
1
TenYearCHD


glucose vs TenYearCHD
glucose
100
90
80
70
60
50
0
1
TenYearCHD

2. Categorical Features vs Target

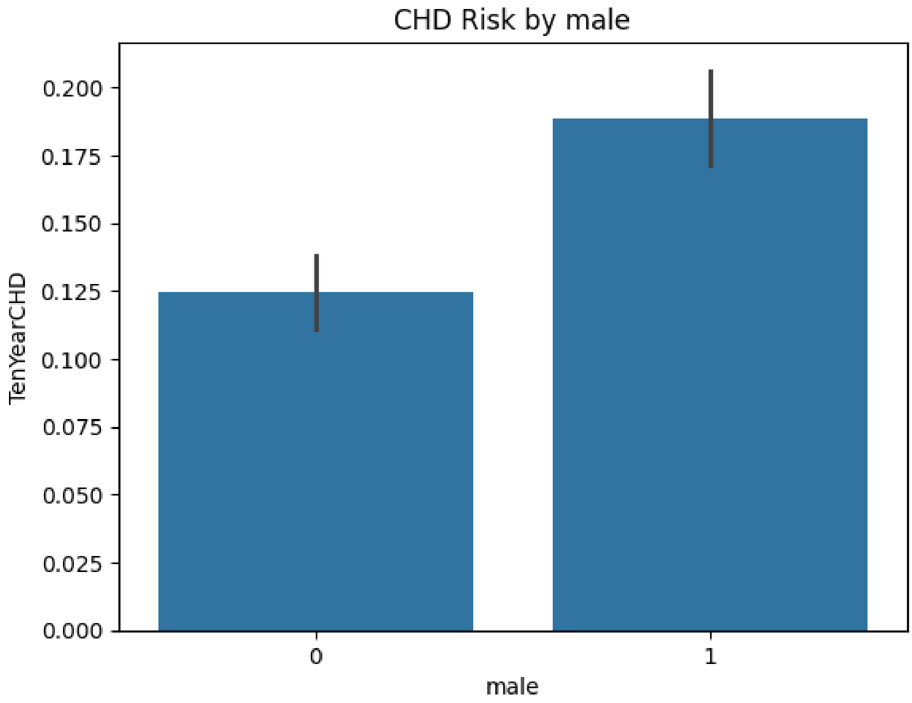
main.ipynb M

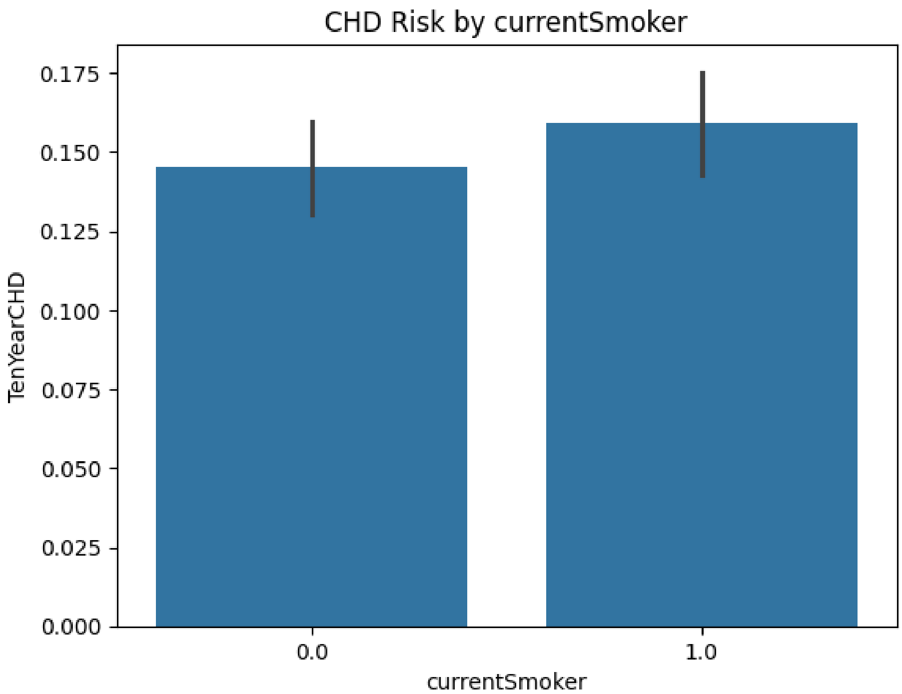
Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ...

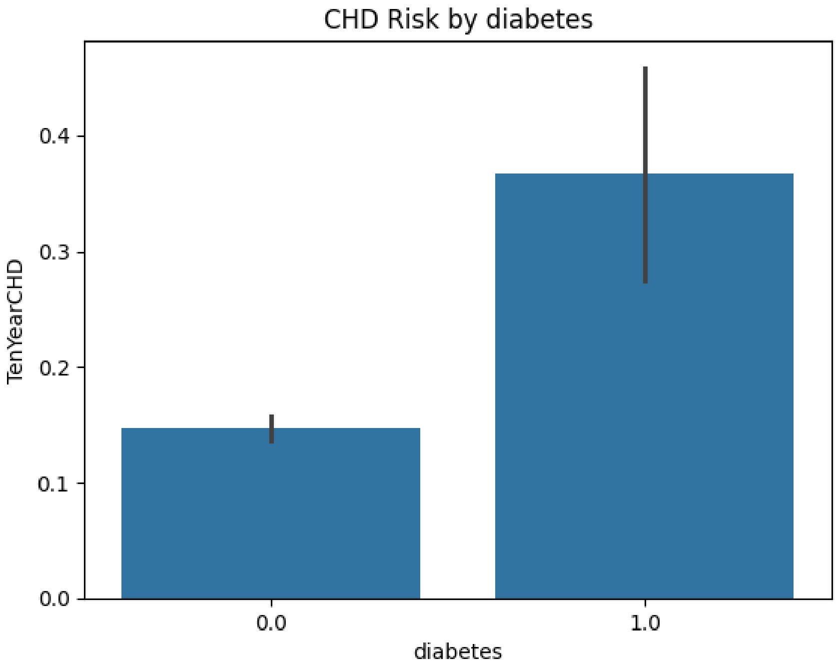
Categorical Feature vs Target

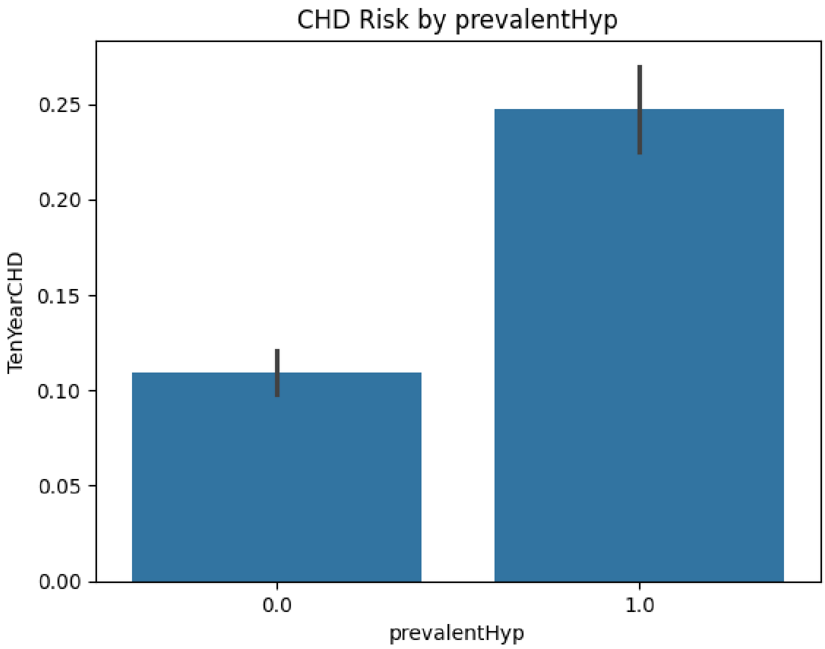
```python
cat_cols = ['male', 'currentSmoker', 'diabetes', 'prevalentHyp']

for col in cat_cols:
    sns.barplot(x=col, y='TenYearCHD', data=df,)
    plt.title(f"CHD Risk by {col}")
    plt.show()
```

[60] Python









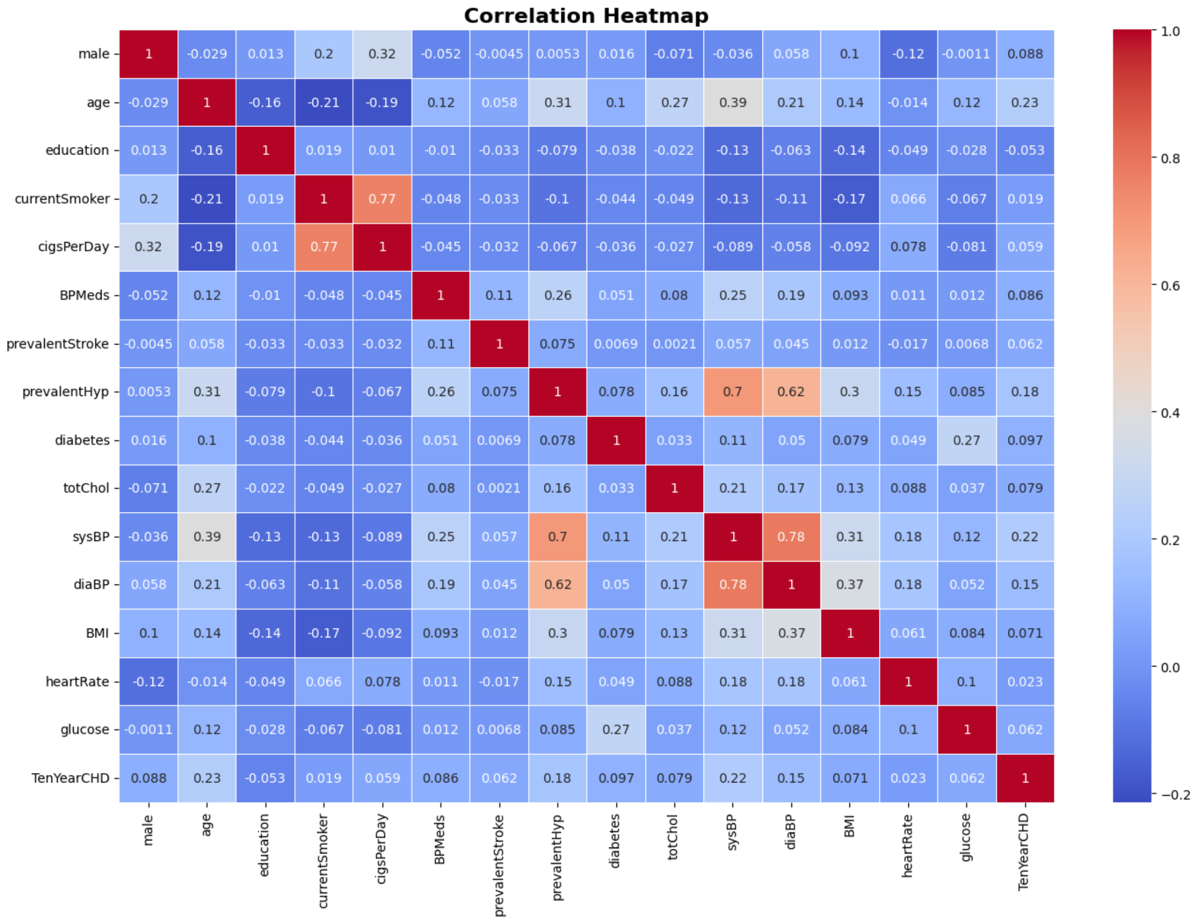
Bivariate Observations

- CHD patients have higher median age and blood pressure
- Diabetes and hypertension show strong association with CHD
- Smoking increases CHD probability

4. Multivariate Analysis

1. Correlation Matrix

```python
plt.figure(figsize=(14, 10))
sns.heatmap(
    df.corr(),
    mask=None,
    cmap='coolwarm',
    linewidths=0.5,
    annot=True
)
plt.title("Correlation Heatmap", fontsize=16, fontweight='bold')
plt.tight_layout()
plt.show()
```



Multivariate Observations

- Strong correlation between sysBP and diaBP
- BMI and glucose are positively correlated
- Multicollinearity exists and affects linear models

2. Visualizing distributions

Gokulnaath M 24BAD028 - Jayaraksha Reguraj 24BAD044

1. KDE Distribution by Target

main.ipynb M

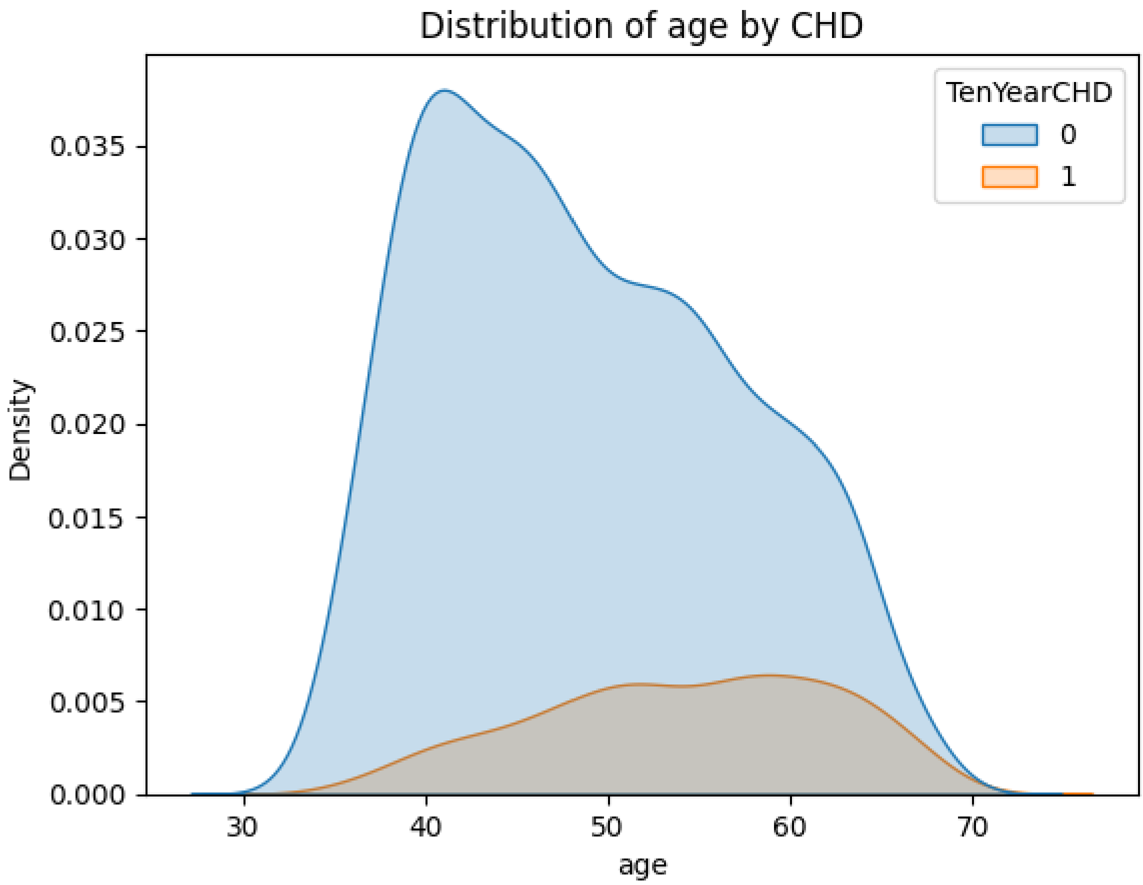
Generate + Code + Markdown | Interrupt Restart Clear All Outputs Go To | Jupyter Variables Outline ... Python 3.13.7

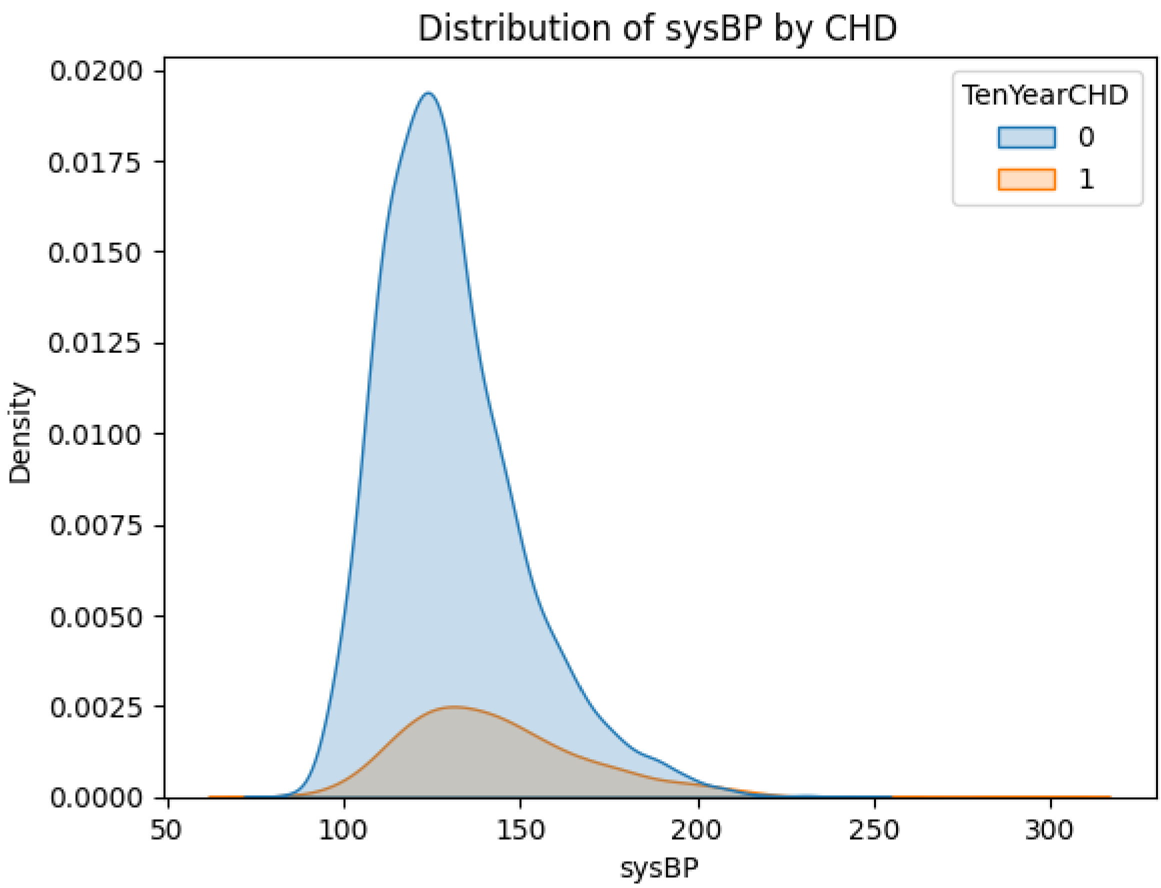
Distribution Analysis

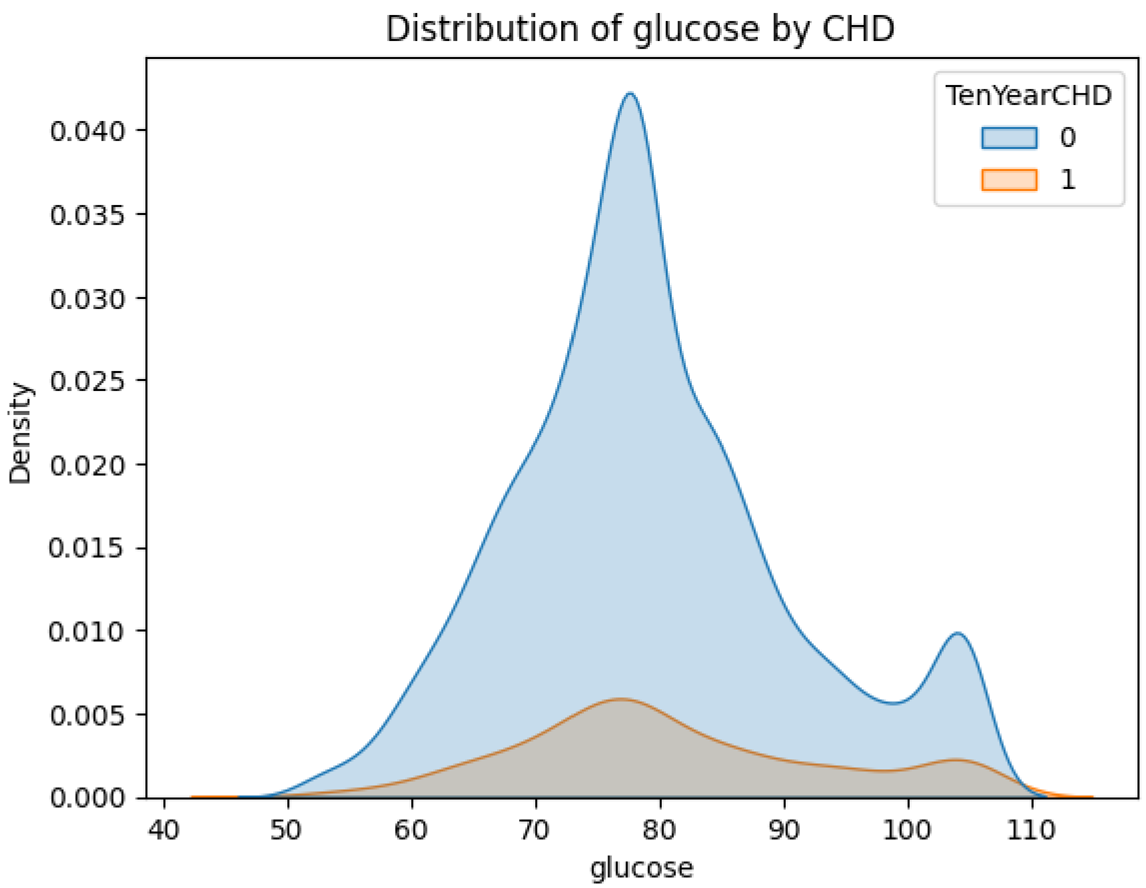
KDE Distribution by Target

```python
for col in ['age', 'sysBP', 'glucose']:
    sns.kdeplot(data=df, x=col, hue='TenYearCHD', fill=True)
    plt.title(f"Distribution of {col} by CHD")
    plt.show()
```

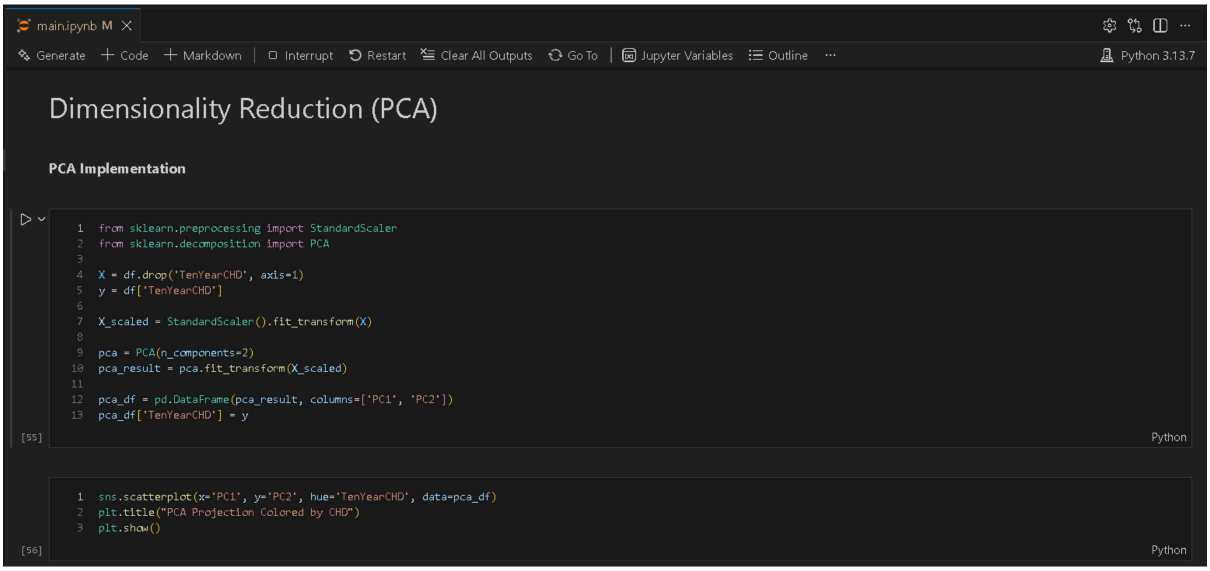
[54] Python




Distribution of sysBP by CHD
TenYearCHD
0
1
Density
0.0200
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
50
100
150
200
250
300
sysBP


Distribution of glucose by CHD
TenYearCHD
0
1
Density
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
40
50
60
70
80
90
100
110
glucose

2. PCA Distribution by Target

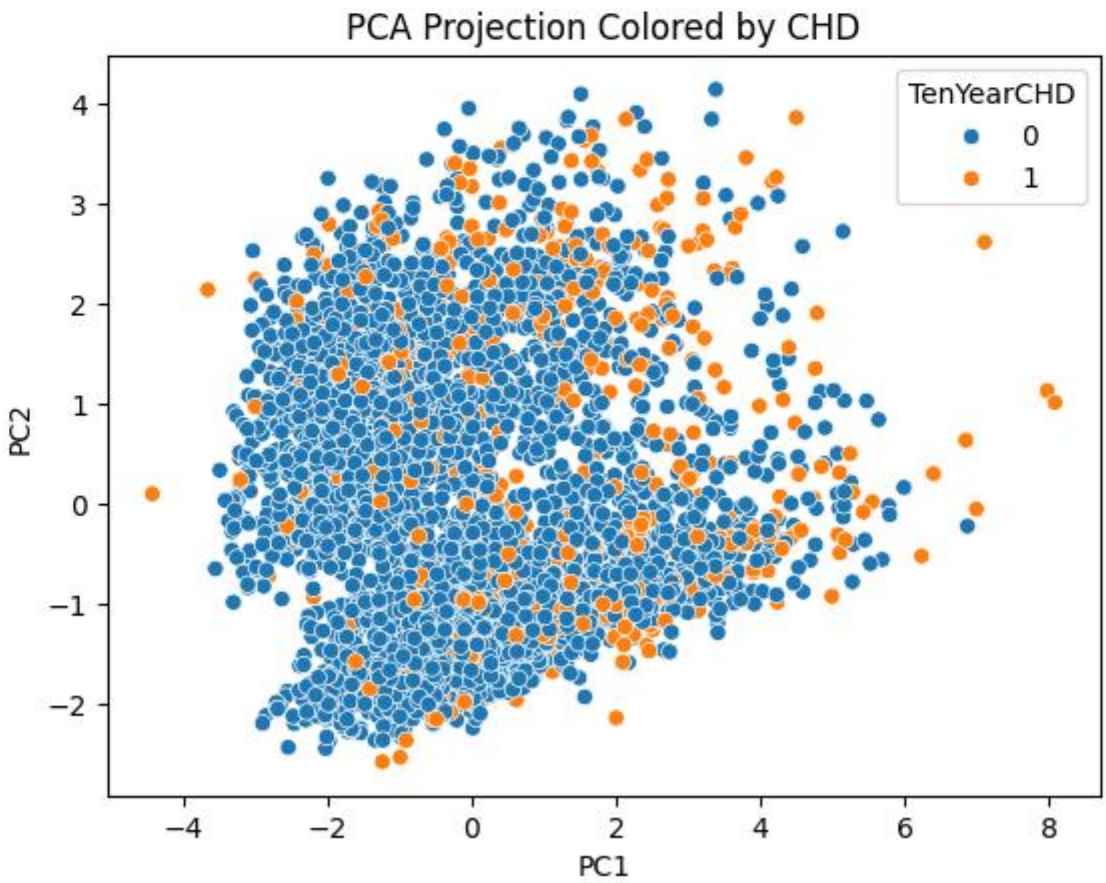

main.ipynb M
Generate + Code + Markdown Interrupt Restart Clear All Outputs Go To Jupyter Variables Outline
Python 3.13.7

Dimensionality Reduction (PCA)

PCA Implementation

```python
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

X = df.drop('TenYearCHD', axis=1)
y = df['TenYearCHD']

X_scaled = StandardScaler().fit_transform(X)

pca = PCA(n_components=2)
pca_result = pca.fit_transform(X_scaled)

pca_df = pd.DataFrame(pca_result, columns=['PC1', 'PC2'])
pca_df['TenYearCHD'] = y
```
[55] Python

```python
sns.scatterplot(x='PC1', y='PC2', hue='TenYearCHD', data=pca_df)
plt.title("PCA Projection Colored by CHD")
plt.show()
```
[56] Python



Resources and References

- **Dataset**
  - https://www.kaggle.com/datasets/dileep070/heart-disease-prediction-using-logistic-regression

- **GitHub**
  - https://github.com/Gokulnaath-gif/24ADI204_DSV_Team14

- **Resources**
  - https://www.geeksforgeeks.org/data-analysis/what-is-exploratory-data-analysis/
  - https://www.ibm.com/think/topics/exploratory-data-analysis
  - https://www.kaggle.com/code/imoore/intro-to-exploratory-data-analysis-eda-in-python
  - https://x.ai/grok
  - https://openai.com/chatgpt
  - https://google.com/gemini