

✓ Practical Machine Learning and Data Exploration

Project-1: Data Preprocessing and Visualization

Student Name: Moayad Khayat

Course: Practical Machine Learning - First Trimester 2026

University: Midocean University

Introduction

This project focuses on the fundamental steps of data preprocessing and exploration, which are essential for building robust machine learning models. Based on the guidelines from **Lecture 4**, we perform the following tasks:

1. **Handling Missing Values:** Identification and imputation of data gaps in the Chronic Kidney Disease dataset.
2. **Handling Outliers:** Identification and removal of extreme values in the Heart Disease dataset using the IQR method.
3. **Data Visualization:** Exploratory Data Analysis (EDA) using various plots to understand feature distributions and correlations.

✓ 0. Import Libraries and Load Data

We import standard libraries and load the datasets specified for this project.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder

# Set project aesthetics
sns.set(style="whitegrid")
import warnings
warnings.filterwarnings("ignore")

# Load datasets
df_kidney = pd.read_csv('kidney_disease.csv')
df_heart = pd.read_csv('heart.csv')

print("Datasets loaded successfully.")
```

Datasets loaded successfully.

✓ 1. Handle Missing Values (Kidney Disease Dataset)

The Chronic Kidney Disease dataset is known for having significant missing values across various clinical features.

```
# 1.1 Detection: Calculate percentage of missing values
missing_val_counts = df_kidney.isnull().sum()
missing_percentage = (missing_val_counts / len(df_kidney)) * 100

print("Columns with Missing Values and their Percentages:")
print(missing_percentage[missing_percentage > 0].sort_values(ascending=False))
```

```
Columns with Missing Values and their Percentages:
rbc      37.939698
rc       32.412060
wc       26.381910
pot      22.110553
sod      21.859296
pcv      17.336683
pc       16.080402
hemo     12.814070
su       12.060302
sg       11.557789
al       11.306533
bgr      11.055276
bu       4.773869
sc       4.271357
bp       3.015075
age      2.261307
pcc      1.005025
ba       1.005025
htn      0.502513
dm       0.502513
cad      0.502513
```

```
appet    0.251256
pe       0.251256
ane      0.251256
dtype: float64
```

Threshold-based Dropping

According to Lecture 4, we drop columns that have more than **30% missing values**.

```
# Drop columns with > 30% missing data
threshold = 30
cols_to_drop = missing_percentage[missing_percentage > threshold].index.tolist()
df_kidney_reduced = df_kidney.drop(columns=cols_to_drop)

print(f"Dropped columns: {cols_to_drop}")
print(f"Remaining columns: {df_kidney_reduced.shape[1]}")
```

```
Dropped columns: ['rbc', 'rc']
Remaining columns: 23
```

Imputation Strategy

We fill missing numerical values with the **Mean** and categorical values with the **Mode**.

```
# Imputation: Mean for numerical, Mode for categorical
for col in df_kidney_reduced.columns:
    if df_kidney_reduced[col].dtype == 'object':
        # Categorical: Fill with Mode
        if not df_kidney_reduced[col].mode().empty:
            df_kidney_reduced[col].fillna(df_kidney_reduced[col].mode()[0], inplace=True)
    else:
        # Numerical: Fill with Mean
        df_kidney_reduced[col].fillna(df_kidney_reduced[col].mean(), inplace=True)

print("Missing values after imputation:", df_kidney_reduced.isnull().sum().sum())
```

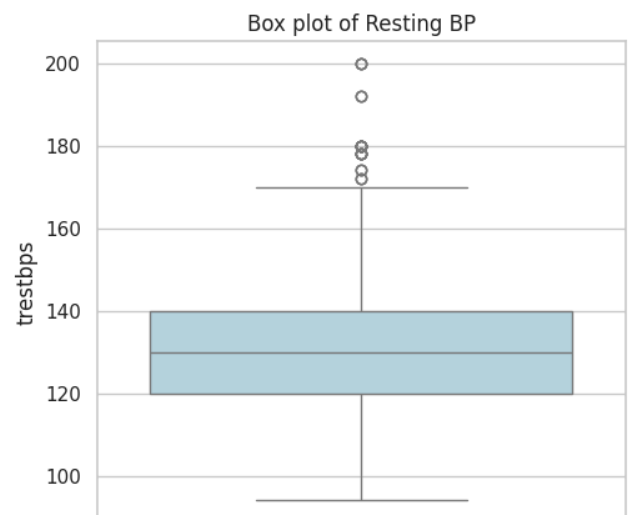
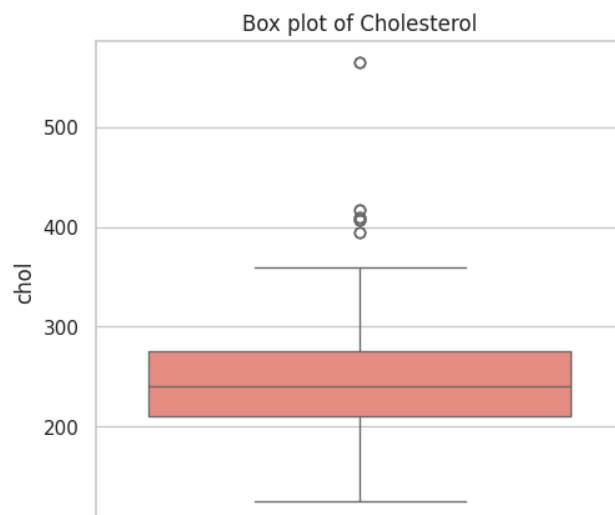
```
Missing values after imputation: 0
```

2. Handle Outliers (Heart Disease Dataset)

We use the Heart Disease dataset to demonstrate outlier detection using **Box plots** and the **IQR method**.

```
# Visualize outliers in the Heart dataset (Cholesterol and Resting Blood Pressure)
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
sns.boxplot(y=df_heart['chol'], color='salmon')
plt.title('Box plot of Cholesterol')

plt.subplot(1, 2, 2)
sns.boxplot(y=df_heart['trestbps'], color='lightblue')
plt.title('Box plot of Resting BP')
plt.show()
```



✓ IQR Filter

We effectively remove data points that lie beyond $1.5 \times IQR$ from the quartiles.

```
def remove_outliers_iqr(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    return df[(df[column] >= lower_bound) & (df[column] <= upper_bound)]

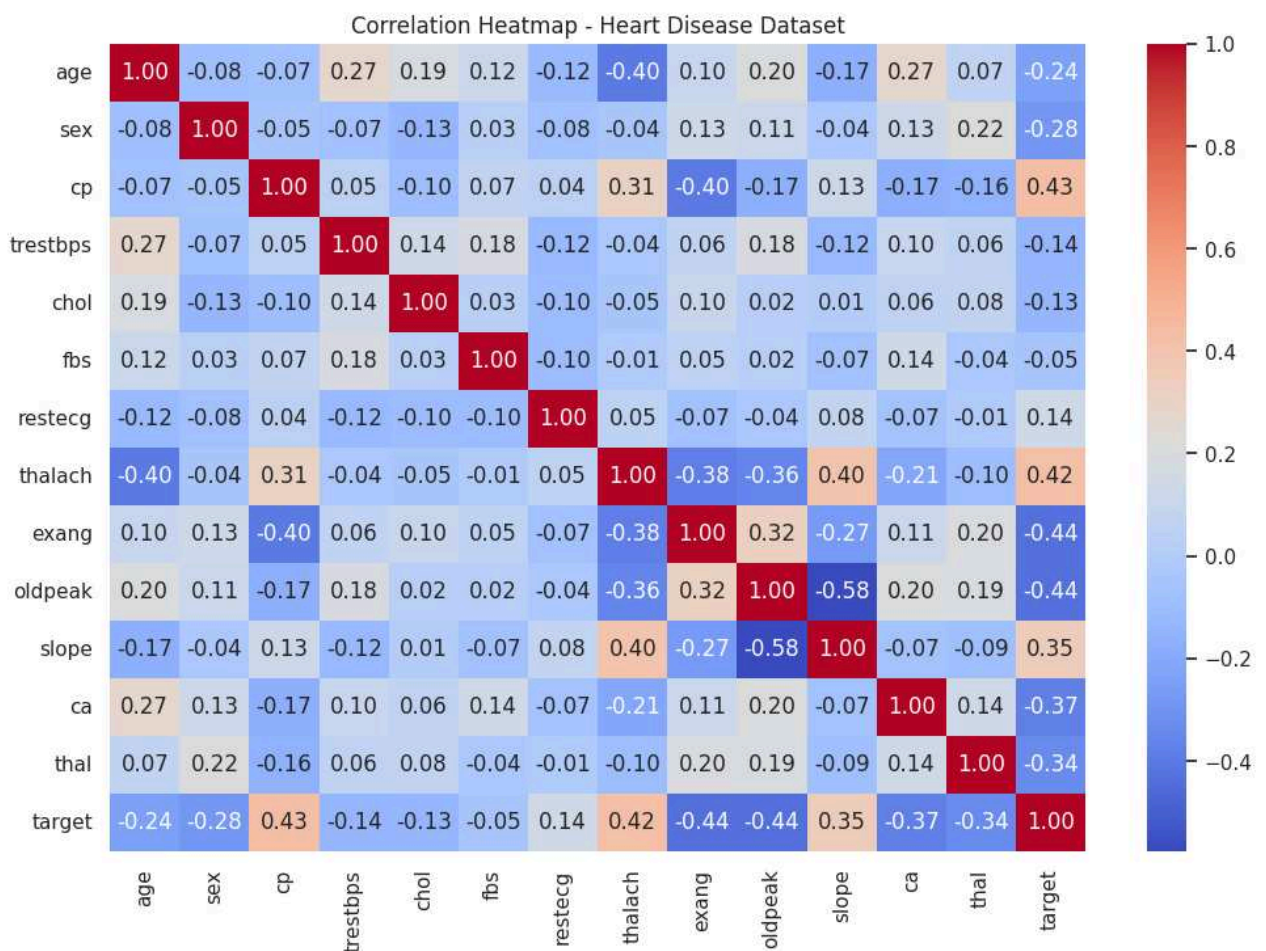
# Removing outliers from 'chol' (Cholesterol)
df_heart_clean = remove_outliers_iqr(df_heart, 'chol')
print(f"Heart dataset rows before: {len(df_heart)}")
print(f"Heart dataset rows after removing 'chol' outliers: {len(df_heart_clean)}")
```

```
Heart dataset rows before: 1025
Heart dataset rows after removing 'chol' outliers: 1009
```

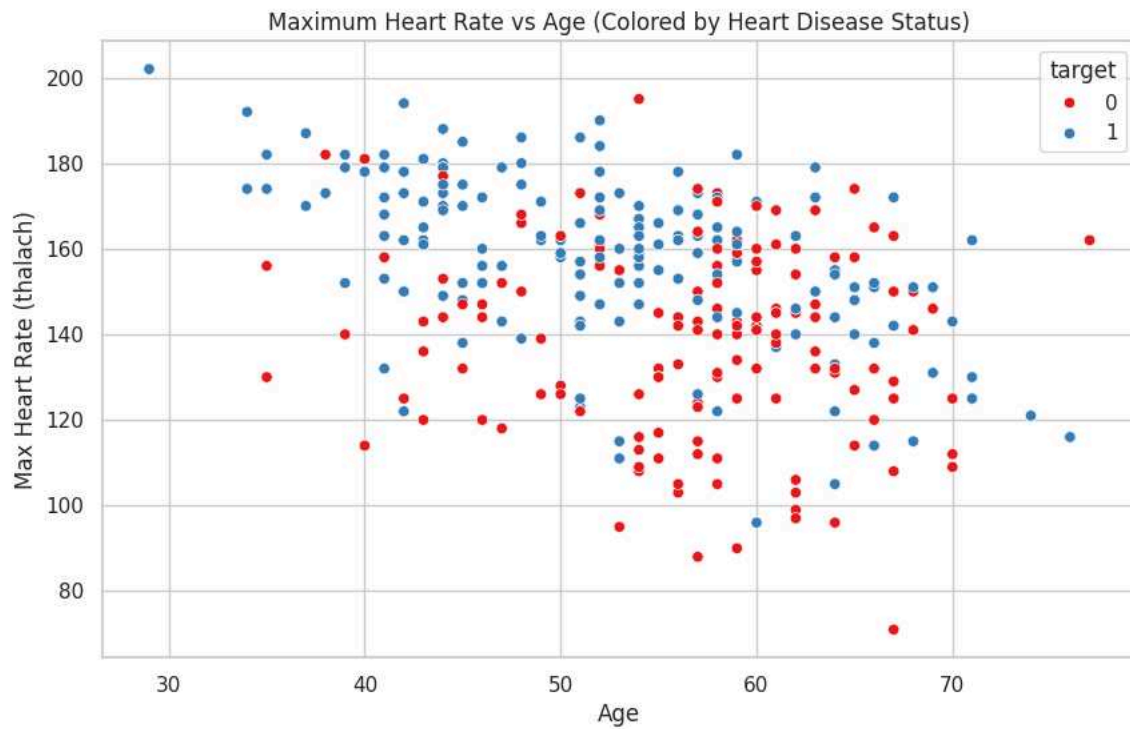
✓ 3. Data Visualization

We perform Exploratory Data Analysis (EDA) as demonstrated in the lecture slides.

```
# 3.1 Correlation Heatmap (Heart Dataset)
plt.figure(figsize=(12, 8))
sns.heatmap(df_heart_clean.corr(), annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap - Heart Disease Dataset')
plt.show()
```

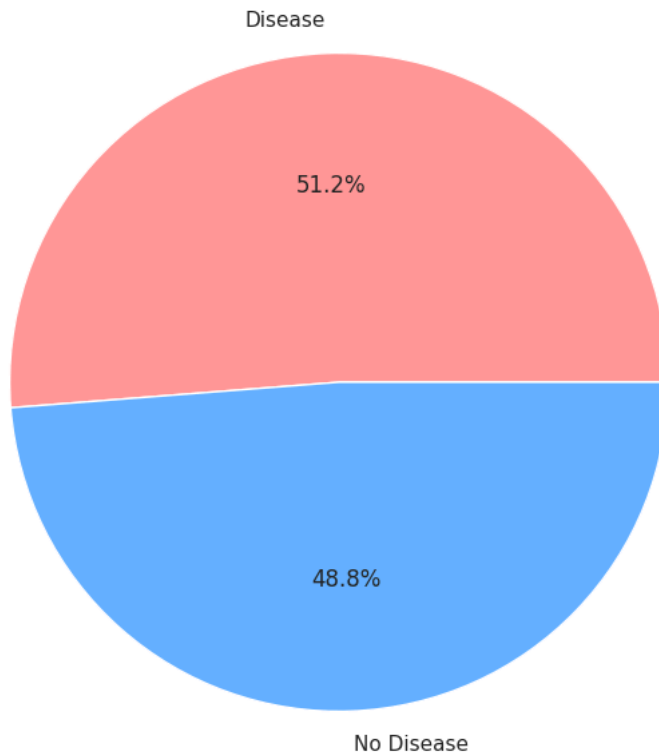


```
# 3.2 Scatter Plot: Max Heart Rate vs. Age (Colored by Target Outcome)
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df_heart_clean, x='age', y='thalach', hue='target', palette='Set1')
plt.title('Maximum Heart Rate vs Age (Colored by Heart Disease Status)')
plt.xlabel('Age')
plt.ylabel('Max Heart Rate (thalach)')
plt.show()
```

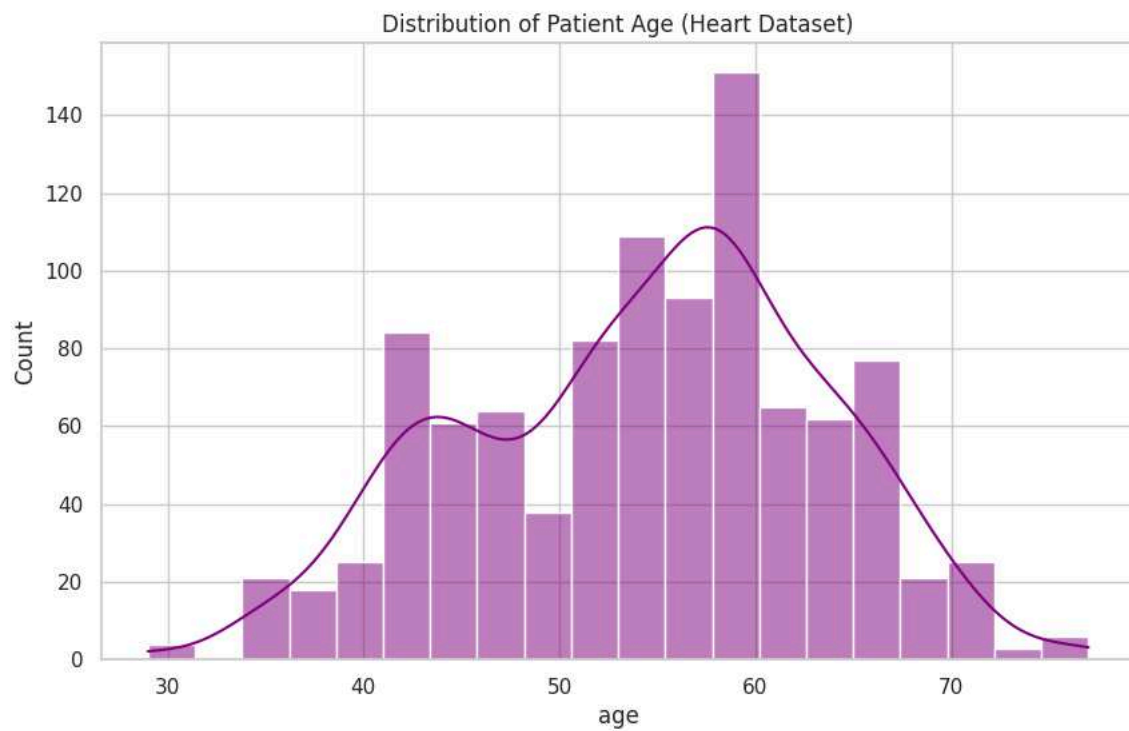


```
# 3.3 Pie Chart: Target Distribution (Heart Dataset)
counts = df_heart_clean['target'].value_counts()
plt.figure(figsize=(8, 8))
plt.pie(counts, labels=['Disease', 'No Disease'], autopct='%1.1f%%', colors=['#ff9999', '#66b3ff'])
plt.title('Distribution of Target Classes in Heart Dataset')
plt.show()
```

Distribution of Target Classes in Heart Dataset



```
# 3.4 Histogram of Age Distribution
plt.figure(figsize=(10, 6))
sns.histplot(df_heart_clean['age'], bins=20, kde=True, color='purple')
plt.title('Distribution of Patient Age (Heart Dataset)')
plt.show()
```



Conclusion

In this project, we applied the specific data preprocessing methods taught in Lecture 4. We successfully handled missing values in the Kidney Disease dataset by dropping columns above the 30% threshold and imputing remaining values with Mean/Mode. We also handled outliers in the Heart Disease dataset using the IQR method. Finally, through various visualizations, we explored the demographic and clinical relationships within the health data, providing a solid foundation for further predictive modeling.