KIET GROUP OF INSTITUTIONS

11 MARCH 2025

HEALTHCARE DATA CLEANING

PRESENTED BY
SHOURYA GARG

ROLL NO.
18 202401100300238

CLASS
CSEAI - D

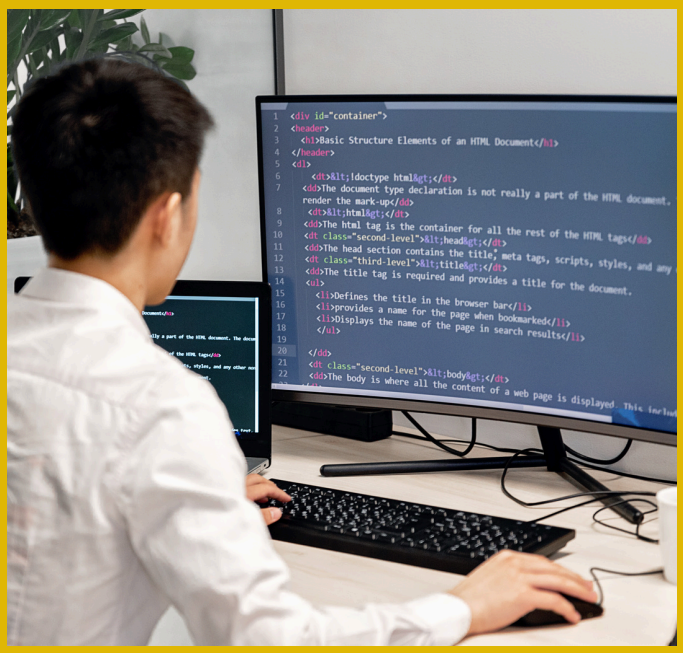
TABLE OF CONTENTS

INTRODUCTION

In this project, we are working with a healthcare dataset that contains information about patients. The dataset includes details like Age, Blood Pressure, Sugar Level, and Weight. The main goal of the project is to clean the data, calculate the Body Mass Index (BMI) for each patient, and then categorize them based on their BMI.

BMI is a number that helps in understanding if a person has a healthy weight. We will divide patients into different categories like Underweight, Normal weight, Overweight, and Obese based on their BMI.

After cleaning the data, we will use a bar graph to show how many patients fall into each BMI category. This helps us understand the overall health of the patients in the dataset.



METHODOLOGY

1) Data Collection:

The data was provided in a CSV file that contains information about patients, including their Age, Blood Pressure, Sugar Level, and Weight.

2) Data Cleaning:

- **Handling Missing Data:** If any data was missing in the columns (like Age, Blood Pressure, Sugar Level, or Weight), we filled it with the average (median) value of that column.
- **Removing Duplicates:** If there were any duplicate rows in the dataset, we removed them to avoid repeating data.
- **Handling Outliers:** We removed any patient data where the Age was unrealistic (greater than 100 years) to keep the data accurate.

3) BMI Calculation:

We calculated the Body Mass Index (BMI) for each patient using their Weight and an assumed average height of 1.7 meters.

The formula for BMI is:

$$\text{BMI} = \frac{\text{WEIGHT(KG)}}{\text{HEIGHT(M)}^2}$$

4) BMI Categorization:

Once the BMI was calculated, we classified each patient into one of the following categories:

- Underweight: BMI < 18.5
- Normal weight: 18.5 ≤ BMI < 24.9
- Overweight: 25 ≤ BMI < 29.9
- Obese: BMI ≥ 30

5) Data Visualization:

We created a bar graph to show how many patients fall into each BMI category. This gives a quick visual representation of the data.

CODE

```
import pandas as pd
from google.colab import drive
from prettytable import PrettyTable
import matplotlib.pyplot as plt
import seaborn as sns

# Step 1: Mount Google Drive
drive.mount('/content/drive')

# Step 2: Define the file path and load the CSV file from Google Drive
file_path = '/content/drive/MyDrive/healthcare_data.csv' # Path to your CSV file in
Google Drive
df = pd.read_csv(file_path)

# Step 3: Clean the column names by stripping spaces or special characters
df.columns = df.columns.str.strip()

# Step 4: Assuming average height of 1.7 meters for BMI calculation
average_height = 1.7 # in meters

# Step 5: BMI Calculation
df['BMI'] = df['Weight'] / (average_height ** 2)

# Step 6: BMI Category Classification
def categorize_bmi(bmi):
  if bmi < 18.5:
    return 'Underweight'
  elif 18.5 <= bmi < 24.9:
    return 'Normal weight'
  elif 25 <= bmi < 29.9:
    return 'Overweight'
  else:
    return 'Obese'

df['BMICategory'] = df['BMI'].apply(categorize_bmi)
```

```
# Step 7: Display the cleaned data (all rows)
print("Data Cleaning Complete!")
print("The cleaned data is ready and saved in Google Drive.")
print("-" * 50)

# Show all rows of the cleaned dataset (with BMI and BMICategory)
print("All rows of the cleaned data with BMI and BMI Category:")

# Create a PrettyTable instance (convert columns to list)
cleaned_table = PrettyTable(df.columns.tolist())

# Add all rows to the cleaned data table
for index, row in df.iterrows():
  cleaned_table.add_row(row)

print(cleaned_table)
print("-" * 50)

# Step 8: Check for missing values in each column
print("Missing Values Summary:")
missing_values = df.isnull().sum()

# Create a PrettyTable for missing values
missing_table = PrettyTable()
missing_table.field_names = ["Column", "Missing Values"]

# Add rows to the missing values table
for column, count in missing_values.items():
  missing_table.add_row([column, count])

print(missing_table)
print("-" * 50)

# Step 9: Fill missing values with the median (if any)
df['Age'] = df['Age'].fillna(df['Age'].median())
df['BloodPressure'] = df['BloodPressure'].fillna(df['BloodPressure'].median())
df['SugarLevel'] = df['SugarLevel'].fillna(df['SugarLevel'].median())
df['Weight'] = df['Weight'].fillna(df['Weight'].median())

# Step 10: Remove duplicate rows from the dataset
df_cleaned = df.drop_duplicates()

# Step 11: Check data types and convert them if necessary
df_cleaned['PatientID'] = df_cleaned['PatientID'].astype(str) # Ensure 'PatientID' is a
string
df_cleaned['Age'] = pd.to_numeric(df_cleaned['Age'], errors='coerce')
df_cleaned['BloodPressure'] = pd.to_numeric(df_cleaned['BloodPressure'],
errors='coerce')
df_cleaned['SugarLevel'] = pd.to_numeric(df_cleaned['SugarLevel'], errors='coerce')
df_cleaned['Weight'] = pd.to_numeric(df_cleaned['Weight'], errors='coerce')
```

```
# Step 12: Remove unrealistic outliers (e.g., Age > 100)
df_cleaned = df_cleaned[df_cleaned['Age'] <= 100]

# Step 13: Show the cleaned dataset (again, after all cleaning steps)
print("All rows of the cleaned data after all cleaning steps:")

# Create a PrettyTable for cleaned data (convert columns to list)
final_table = PrettyTable(df_cleaned.columns.tolist())

# Add all rows to the final cleaned data table
for index, row in df_cleaned.iterrows():
  final_table.add_row(row)

print(final_table)
print("-" * 50)

# Step 14: Show missing values in cleaned data (should be 0 for all columns)
print("Missing Values in Cleaned Data:")

# Create a PrettyTable for missing values in cleaned data
missing_cleaned_table = PrettyTable()
missing_cleaned_table.field_names = ["Column", "Missing Values"]

missing_values_cleaned = df_cleaned.isnull().sum()

# Add rows to the cleaned missing values table
for column, count in missing_values_cleaned.items():
  missing_cleaned_table.add_row([column, count])

print(missing_cleaned_table)
print("-" * 50)

# Step 15: Create a summary table showing the number of people in each BMI
category
print("Summary of BMI Categories:")

# Count the number of people in each BMI category
bmi_summary = df_cleaned['BMICategory'].value_counts().reset_index()
bmi_summary.columns = ['BMI Category', 'Number of People']

# Create a PrettyTable to display the summary
bmi_summary_table = PrettyTable(bmi_summary.columns.tolist())

# Add all rows to the summary table
for index, row in bmi_summary.iterrows():
  bmi_summary_table.add_row(row)

print(bmi_summary_table)
print("-" * 50)
```

```
# Step 16: Create a Bar Graph for BMI Categories
plt.figure(figsize=(8, 6))
sns.countplot(x='BMICategory', data=df_cleaned, palette='Set2')

# Title and labels
plt.title("Distribution of BMI Categories", fontsize=14)
plt.xlabel("BMI Category", fontsize=12)
plt.ylabel("Number of People", fontsize=12)

# Show the plot
plt.show()

# Step 17: Save the cleaned data back to Google Drive
output_path = '/content/drive/MyDrive/cleaned_healthcare_data_with_bmi.csv' # Path to save the cleaned data
df_cleaned.to_csv(output_path, index=False)

# Step 18: Inform user where the cleaned data is saved
print("Cleaned data with BMI and BMI Category has been saved to Google Drive as:")
print(output_path)
```

SNAPSHOTS OF OUTPUT

All rows of the cleaned data after all cleaning steps:

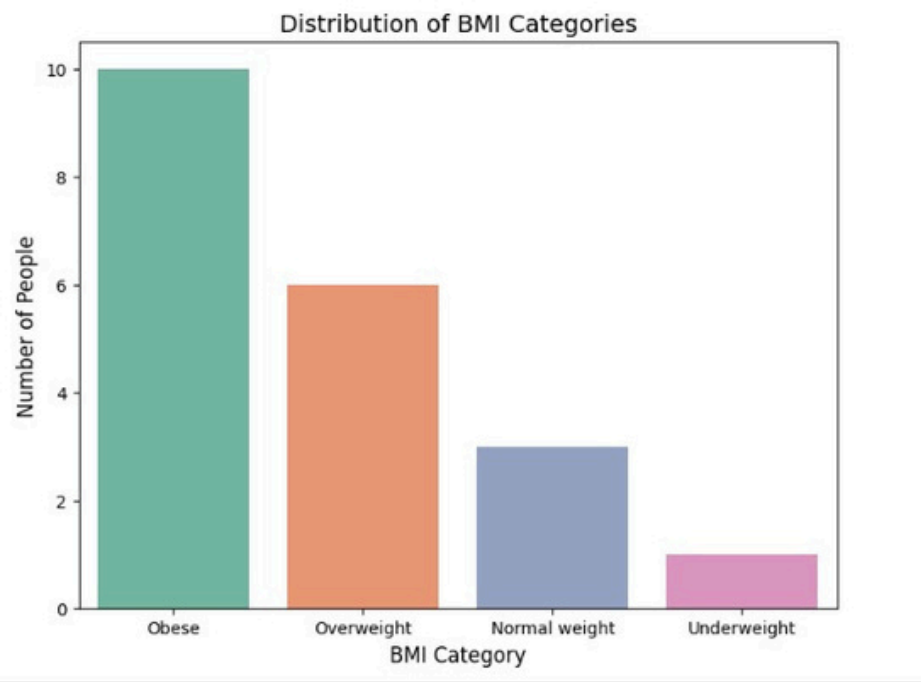
PatientID	Age	BloodPressure	SugarLevel	Weight	BMI	BMICategory
1	44	118	87.89249491582838	105.56803408361246	36.528731516820926	Obese
2	39	109	177.32180303065556	105.70342555971732	36.575579778448905	Obese
3	49	149	144.14827323225148	77.78706964302448	26.915941052949652	Overweight
4	58	121	90.35540377032444	115.24478387577324	39.87708784628832	Obese
5	35	109	126.42179998391184	70.38379045448772	24.3542527524179	Normal weight
6	25	129	95.27311377325915	119.050356353696	41.19389493207475	Obese
7	46	132	146.60771849705907	62.17751535920798	21.51471119695778	Normal weight
8	28	93	109.7549861802895	81.79225909369181	28.301819755602708	Overweight
9	60	145	103.1938308024857	94.6373684790234	32.7464942833991	Obese
10	55	125	197.726355774952	118.59398079624158	41.03597951427045	Obese
11	41	143	180.57879611740577	103.58465512371292	35.84244121927783	Obese
12	48	141	181.97250706947565	61.45498222950457	21.26469973339259	Normal weight
13	58	93	181.7836075144378	50.68483483788439	17.538005134216053	Underweight
14	35	145	133.3857116888367	113.18663218872322	39.164924632776206	Obese
15	67	176	87.00502726461762	84.9385760131818	29.390510731204778	Overweight
16	70	109	193.2727707077015	77.71503785899397	26.89101656020553	Overweight
17	43	148	135.9394820705097	106.57598882252016	36.87750478287895	Obese
18	74	122	129.41123366125004	83.3004255284348	28.82367665343765	Overweight
19	19	147	125.48395754434272	74.08193838763104	25.63388871544327	Overweight
20	56	119	160.71585302410716	111.86569750294632	38.707853807247865	Obese

Missing Values in Cleaned Data:

Column	Missing Values
PatientID	0
Age	0
BloodPressure	0
SugarLevel	0
Weight	0
BMI	0
BMICategory	0

Summary of BMI Categories:

BMI Category	Number of People
Obese	10
Overweight	6
Normal weight	3
Underweight	1



REFERENCES

www.canva.com

www.wikipedia.com

www.google.com