A PRELIMINARY MINI PROJECT REPORT ON

Sleep Disorder Prediction

SUBMITTED TOWARDS THE PARTIAL FULFILLMENT OF THE REQUIREMENTS OF

BACHELOR OF ENGINEERING (S.Y. B. Tech.)

Academic Year: 2024-25

By:

Vaishnavi B. Gaikwad (123B5B294)

Under The Guidance of

Prof. Namrata Gawande

DEPARTMENT OF COMPUTER ENGINEERING,

PIMPRI CHINCHWAD COLLEGE OF

ENGINEERING SECTOR 26, NIGDI,

PRADHIKARAN

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING,

SECTOR 26, NIGDI, PRADHIKARAN

CERTIFICATE

This is to certify that, the project entitled

"SLEEP DISORDER PREDICTION"

is successfully carried out as a **Skill Development Laboratory I** mini project and successfully submitted by following students of "PCET's Pimpri Chinchwad College of Engineering,

Nigdi, Pune-44".

Under the guidance of

Mrs. Namrata Gawande

In the partial fulfillment of the requirements for the

S.Y. B. Tech. (Computer Engineering)

Vaishnavi Gaikwad(123B5B294)

Mrs. Namrata Gawande

Project Guide

INDEX

1. Introduction

a) Background

Sleep health is a critical component of well-being, influenced by various lifestyle factors such as physical activity, stress levels, and daily routines. However, with the rise of modern, often sedentary lifestyles and increased stress, many individuals experience disruptions in sleep, which can lead to long-term health issues like obesity, cardiovascular problems, and psychological stress. This project aims to analyze and derive insights from a dataset containing attributes such as sleep duration, quality of sleep, physical activity, stress levels, and other health indicators. By examining these factors, this research will provide a deeper understanding of how lifestyle choices impact sleep health.

b) Problem Statement

There is a need for data-driven insights to identify the lifestyle factors most significantly affecting sleep quality. This project addresses the question: *What are the key lifestyle factors that influence sleep quality and disorders, and how can they be managed to improve sleep health?*

c) Project Idea/Objectives

The main objectives of this project are:

1. To analyze correlations between sleep duration, sleep quality, and lifestyle factors (such as physical activity, stress levels, BMI, and daily steps).
2. To identify the lifestyle behaviors that contribute to or mitigate the risk of sleep disorders.
3. To visualize relationships between various health indicators (e.g., blood pressure, heart rate) and sleep quality to make insights more accessible.
4. To develop a model that can predict sleep disorder risk based on lifestyle and health metrics.

d) Motivation

With increasing sleep-related health issues globally, this project is motivated by the need for accessible insights into factors influencing sleep quality. The goal is to help individuals make informed lifestyle choices for better sleep health, as well as to provide a resource for healthcare providers who wish to guide patients in sleep-related lifestyle adjustments.

e) Scope

This project will focus on analyzing the sleep health and lifestyle dataset provided. The analysis includes preprocessing the data, identifying correlations between variables, visualizing trends, and developing a predictive model for sleep disorder risks based on lifestyle metrics. However, the scope is limited to the dataset's observational nature, which restricts the project to correlational insights rather than causal inferences. Additionally, the findings may not apply universally, as the dataset might reflect specific population characteristics.

2. Data Collection

Dataset Information

The dataset used for this analysis is titled "Sleep Health and Lifestyle Dataset." It contains data on various lifestyle and health-related attributes that may influence sleep patterns and quality. The dataset is structured with each row representing an individual, and columns represent different attributes related to demographics, health, and sleep quality. This dataset is suitable for exploring the impact of lifestyle factors on sleep health and was chosen due to its variety of features, which can provide insights into the relationships between sleep and other lifestyle aspects.

Dataset Overview:

The Sleep Health and Lifestyle Dataset comprises 400 rows and 13 columns, covering a wide range of variables related to sleep and daily habits. It includes details such as gender, age, occupation, sleep duration, quality of sleep, physical activity level, stress levels, BMI category, blood pressure, heart rate, daily steps, and the presence or absence of sleep disorders.

Key Features of the Dataset:

Comprehensive Sleep Metrics: Explore sleep duration, quality, and factors influencing sleep patterns.

Lifestyle Factors: Analyze physical activity levels, stress levels, and BMI categories.

Cardiovascular Health: Examine blood pressure and heart rate measurements.

Sleep Disorder Analysis: Identify the occurrence of sleep disorders such as Insomnia and Sleep Apnea.

Dataset Attributes:

Person ID: An identifier for each individual.

Gender: The gender of the person (Male/Female).

Age: The age of the person in years.

Occupation: The occupation or profession of the person.

Sleep Duration (hours): The number of hours the person sleeps per day.

Quality of Sleep (scale: 1-10): A subjective rating of the quality of sleep, ranging from 1 to 10.

Physical Activity Level (minutes/day): The number of minutes the person engages in physical activity daily.

Stress Level (scale: 1-10): A subjective rating of the stress level experienced by the person, ranging from 1 to 10.

BMI Category: The BMI category of the person (e.g., Underweight, Normal, Overweight).

Blood Pressure (systolic/diastolic): The blood pressure measurement of the person, indicated as systolic pressure over diastolic pressure.

Heart Rate (bpm): The resting heart rate of the person in beats per minute.

Daily Steps: The number of steps the person takes per day.

Sleep Disorder: The presence or absence of a sleep disorder in the person (None, Insomnia, Sleep Apnea).

Data Source

The dataset was sourced from www.kaggle.com.

Link to access the dataset:

https://www.kaggle.com/datasets/uom190346a/sleep-health-and-lifestyle-dataset/data

This data is publicly accessible and provides a comprehensive view of factors that may affect sleep quality and health, making it valuable for analysis in health and wellness research.

3. Exploratory Data Analysis (EDA)

- **Data Preprocessing:**Data preprocessing is an important step in the data mining process. It refers to the cleaning, transforming, and integrating of data in order to make it ready for analysis. The goal of data preprocessing is to improve the quality of the data and to make it more suitable for the specific data mining task.

Steps to clean and prepare the dataset (handling missing values, removing outliers, etc.).

PROGRAM and OUTPUTS:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
health =
pd.read_csv("C:\\Users\\vaish\\Downloads\\Sleep_health_and_lifestyle_dataset_new.csv")
print(health)
#print(df.to_string())    ---> prints whole excel sheet
```

```
Person_ID  Gender  Age          Occupation  Sleep_Duration  \
0          1    Male   27    Software Engineer           6.1
1          2    Male   28              Doctor           6.2
2          3    Male   28              Doctor           6.2
3          4    Male   28  Sales Representative           5.9
4          5    Male   28  Sales Representative           5.9
..       ...     ...  ...                 ...           ...
369      370  Female   59               Nurse           8.1
370      371  Female   59               Nurse           8.0
371      372  Female   59               Nurse           8.1
372      373  Female   59               Nurse           8.1
373      374  Female   59               Nurse           8.1

     Quality_of_Sleep  Physical_Activity_Level  Stress_Level BMI_Category  \
0                   6                       42             6   Overweight
1                   6                       60             8       Normal
2                   6                       60             8       Normal
3                   4                       30             8        Obese
4                   4                       30             8        Obese
..                ...                      ...           ...          ...
369                 9                       75             3   Overweight
370                 9                       75             3   Overweight
371                 9                       75             3   Overweight
372                 9                       75             3   Overweight
373                 9                       75             3   Overweight
```

	Blood_Pressure	Heart_Rate	Daily_Steps	Sleep_Disorder
0	126/83	77	4200	No
1	125/80	75	10000	No
2	125/80	75	10000	No
3	140/90	85	3000	Yes
4	140/90	85	3000	Yes
..	...	...	...	...
369	140/95	68	7000	Yes
370	140/95	68	7000	Yes
371	140/95	68	7000	Yes
372	140/95	68	7000	Yes
373	140/95	68	7000	Yes

[374 rows x 13 columns]

type(health)

pandas.core.frame.DataFrame

health.head()

	Person_ID	Gender	Age	Occupation	Sleep_Duration	Quality_of_Sleep	Physical_Activity_Level	Stress_Level	BMI_Category	Blood_Pressure	Heart_Rate	Daily_Steps	Sleep_Disorder
0	1	Male	27	Software Engineer	6.1	6	42	6	Overweight	126/83	77	4200	No
1	2	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10000	No
2	3	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10000	No
3	4	Male	28	Sales Representative	5.9	4	30	8	Obese	140/90	85	3000	Yes
4	5	Male	28	Sales Representative	5.9	4	30	8	Obese	140/90	85	3000	Yes

health.tail()

	Person_ID	Gender	Age	Occupation	Sleep_Duration	Quality_of_Sleep	Physical_Activity_Level	Stress_Level	BMI_Category	Blood_Pressure	Heart_Rate	Daily_Steps	Sleep_Disorder
369	370	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68	7000	Yes
370	371	Female	59	Nurse	8.0	9	75	3	Overweight	140/95	68	7000	Yes
371	372	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68	7000	Yes
372	373	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68	7000	Yes
373	374	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68	7000	Yes

health.shape

(374, 13)

health.describe()

	Person_ID	Age	Sleep_Duration	Quality_of_Sleep	Physical_Activity_Level	Stress_Level	Heart_Rate	Daily_Steps
count	374.000000	374.000000	374.000000	374.000000	374.000000	374.000000	374.000000	374.000000
mean	187.500000	42.184492	7.132086	7.312834	59.171123	5.385027	70.165775	6816.844920
std	108.108742	8.673133	0.795657	1.196956	20.830804	1.774526	4.135676	1617.915679
min	1.000000	27.000000	5.800000	4.000000	30.000000	3.000000	65.000000	3000.000000
25%	94.250000	35.250000	6.400000	6.000000	45.000000	4.000000	68.000000	5600.000000
50%	187.500000	43.000000	7.200000	7.000000	60.000000	5.000000	70.000000	7000.000000
75%	280.750000	50.000000	7.800000	8.000000	75.000000	7.000000	72.000000	8000.000000
max	374.000000	59.000000	8.500000	9.000000	90.000000	8.000000	86.000000	10000.000000

health.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 374 entries, 0 to 373
Data columns (total 13 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Person_ID                374 non-null    int64  
 1   Gender                   374 non-null    object 
 2   Age                      374 non-null    int64  
 3   Occupation               374 non-null    object 
 4   Sleep_Duration           374 non-null    float64
 5   Quality_of_Sleep         374 non-null    int64  
 6   Physical_Activity_Level  374 non-null    int64  
 7   Stress_Level             374 non-null    int64  
 8   BMI_Category             374 non-null    object 
 9   Blood_Pressure           374 non-null    object 
 10  Heart_Rate               374 non-null    int64  
 11  Daily_Steps              374 non-null    int64  
 12  Sleep_Disorder           374 non-null    object 
dtypes: float64(1), int64(7), object(5)
memory usage: 38.1+ KB
```

health.dtypes

```
Person_ID                  int64
Gender                    object
Age                        int64
Occupation                object
Sleep_Duration           float64
Quality_of_Sleep           int64
Physical_Activity_Level    int64
Stress_Level               int64
BMI_Category              object
Blood_Pressure            object
Heart_Rate                 int64
Daily_Steps                int64
Sleep_Disorder            object
dtype: object
```

#check whether any values are null
health.isnull().sum()

```
Person_ID                 0
Gender                    0
Age                       0
Occupation                0
Sleep_Duration            0
Quality_of_Sleep          0
Physical_Activity_Level   0
Stress_Level              0
BMI_Category              0
Blood_Pressure            0
Heart_Rate                0
Daily_Steps               0
Sleep_Disorder            0
dtype: int64
```

health.duplicated()

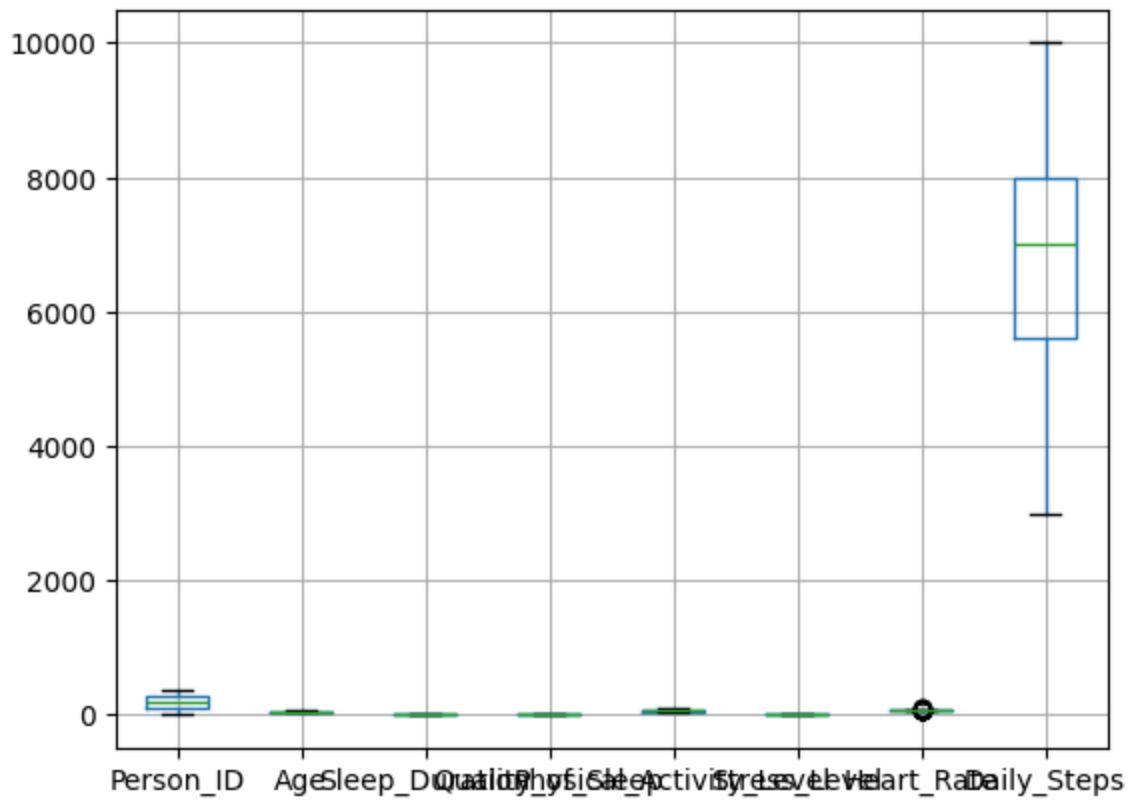
```
0      False
1      False
2      False
3      False
4      False
       ...
369    False
370    False
371    False
372    False
373    False
Length: 374, dtype: bool
```

#Get duplicate records
health[health.duplicated(keep='first')]
print('no. of records before removing duplicates', health.shape[0])
#Remove duplicates
health.drop_duplicates(keep='first', inplace=True)
print('no. of records after removing duplicates', health.shape[0])

no. of records before removing duplicates 374
no. of records after removing duplicates 374

health.boxplot()

<Axes: >



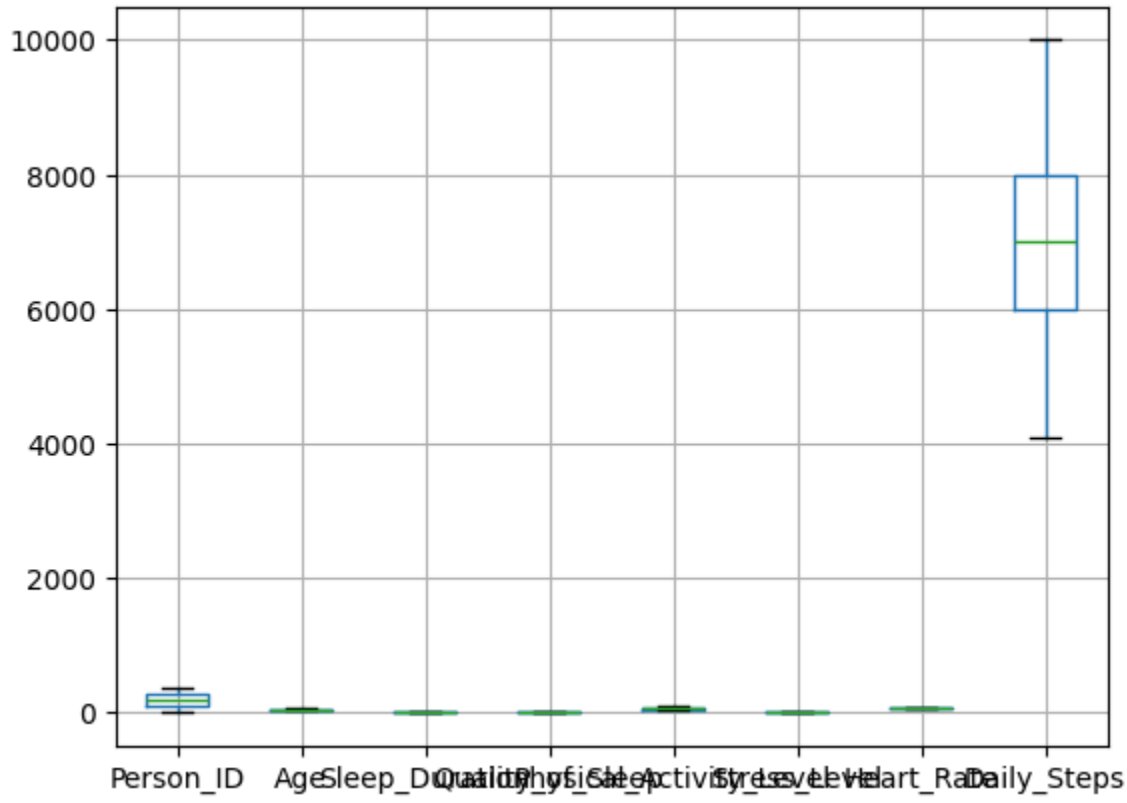

```
def outlier_treatment(datacolumn):
    sorted(datacolumn)
    Q1,Q3 = np.percentile(datacolumn , [25,75])
    IQR = Q3-Q1
    lower_range = Q1 - (1.5 * IQR)
    upper_range = Q3 + (1.5 * IQR)
    return lower_range,upper_range

# Check outliers
l,u=outlier_treatment(health.Sleep_Duration)
health[(health.Sleep_Duration < l) | (health.Sleep_Duration > u)]
print('Before dropping outliers shape of df is: ', health.shape)
# Remove outliers
health.drop(health[(health.Sleep_Duration < l) | (health.Sleep_Duration >
u)].index,inplace=True)
print('After dropping outliers shape of df is: ', health.shape)
```

```
Before dropping outliers shape of df is:  (374, 13)
After dropping outliers shape of df is:  (374, 13)
```

```
# Check outliers
l,u=outlier_treatment(health.Heart_Rate)
health[(health.Heart_Rate < l) | (health.Heart_Rate > u)]
```

```
print('Before dropping outliers shape of df is: ', health.shape)
# Remove outliers
health.drop(health[(health.Heart_Rate < l) | (health.Heart_Rate > u)].index,inplace=True)
print('After dropping outliers shape of df is: ', health.shape)
health.boxplot()
```

```
Before dropping outliers shape of df is:  (374, 13)
After dropping outliers shape of df is:  (359, 13)
<Axes: >
```



- **Data Visualization**: Using graphs like histograms, scatter plots, correlation heatmaps, etc., to explore patterns in the data. Including visualizations of important relationships between variables.

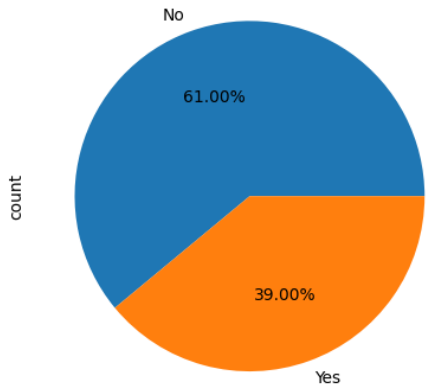
PROGRAM:

```
# Let us see the target variable
M = health.Sleep_Disorder.value_counts()
print(M)
```

```
Sleep_Disorder
No     219
Yes    140
```

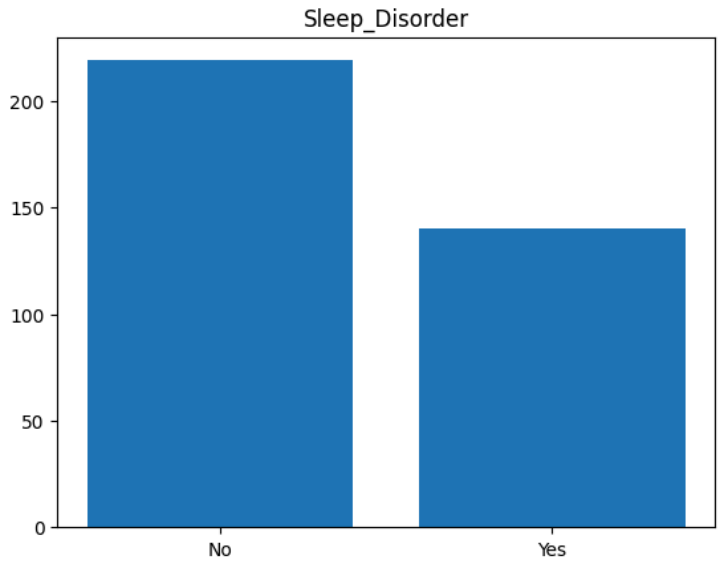
Name: count, dtype: int64

```
M=health.Sleep_Disorder.value_counts().plot(kind='pie',autopct='%1.2f%%')
```

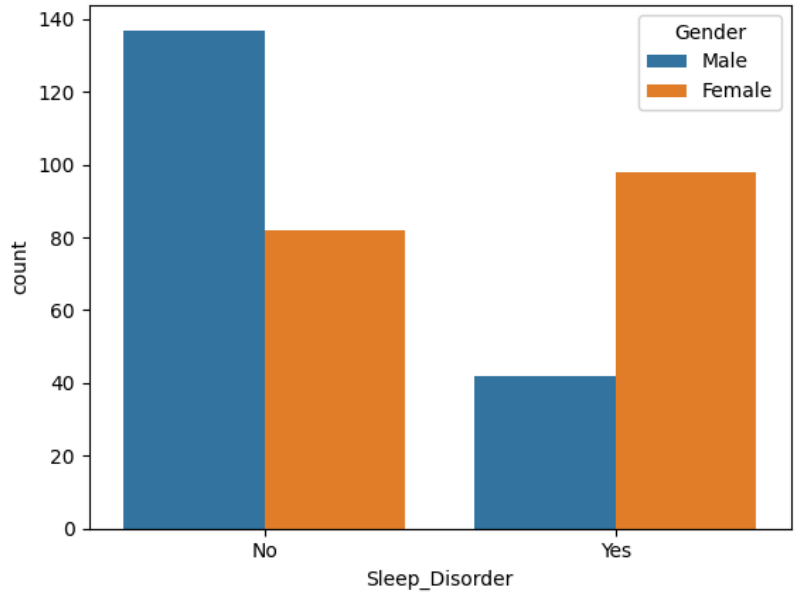



```
M=health.Sleep_Disorder.value_counts()
print(M)
plt.bar(M.index,M.values)
plt.title('Sleep_Disorder')
plt.show()
```

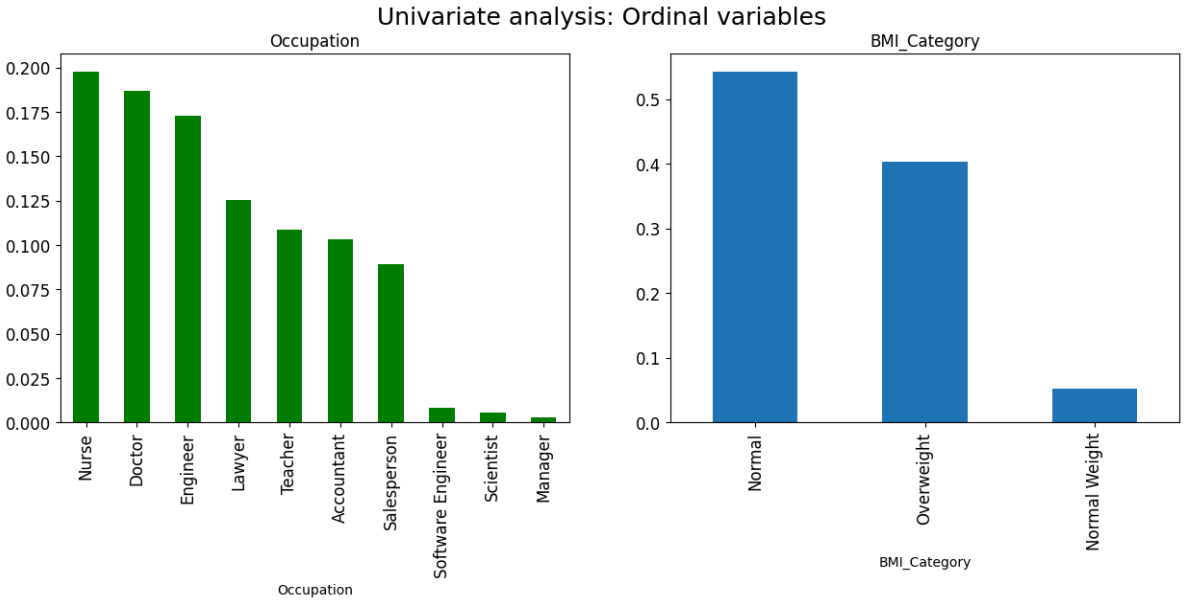
```
Sleep_Disorder
No    219
Yes   140
Name: count, dtype: int64
```



```
sns.countplot(x='Sleep_Disorder', hue='Gender', data=health)
plt.show()
```




```
plt.figure(1)
plt.subplot(121)
health['Occupation'].value_counts(normalize=True).plot.bar(figsize=(15,5), title='Occupation',
fontsize=12, color='green')
plt.subplot(122)
health['BMI_Category'].value_counts(normalize=True).plot.bar(figsize=(15,5),
fontsize=12,title='BMI_Category')
plt.suptitle('Univariate analysis: Ordinal variables', fontsize=18)
plt.show()
```



Explanation:

Bar Plot 1: Occupation

The first bar plot visualizes the distribution of **Occupation** categories in the dataset. The value_counts(normalize=True) function normalizes the values, meaning it shows the relative frequency (proportion) of each category rather than the raw count. Each bar represents a different occupation category, and its height indicates the proportion of individuals in that occupation within the dataset. This plot provides an overview of the occupational distribution in the sample, allowing us to identify the most common and least common occupations.

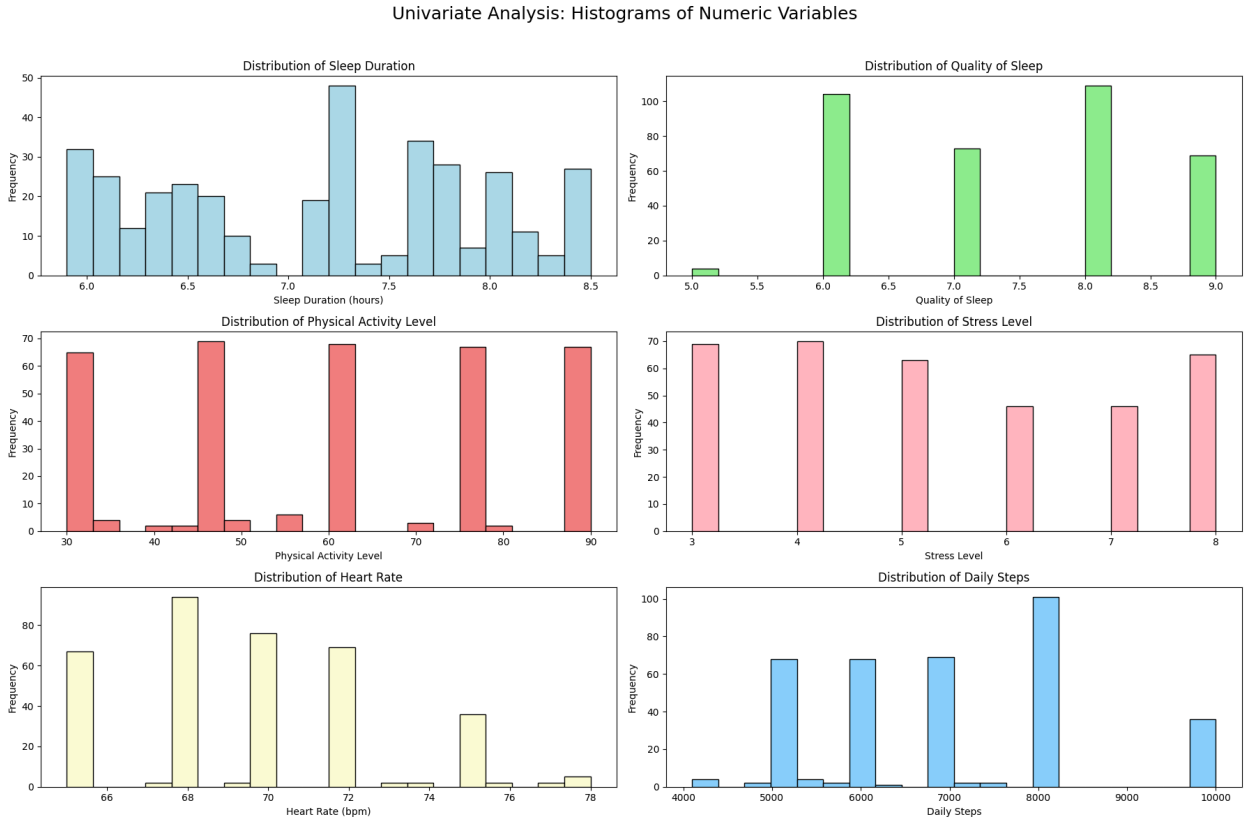
Bar Plot 2: BMI Category

The second bar plot visualizes the distribution of individuals across different **BMI Categories**. Similar to the first plot, it uses value_counts(normalize=True) to display the relative frequency of each BMI category. The categories in this plot likely correspond to standard BMI classifications, such as "Underweight," "Normal," "Overweight," and "Obese." This plot helps in understanding how individuals are distributed across different BMI ranges within the dataset.

PROGRAM:

```
plt.figure(figsize=(18, 12))
# Histograms for each numeric variable
plt.subplot(3, 2, 1)
health['Sleep_Duration'].plot.hist(bins=20, color='lightblue', edgecolor='black')
plt.title('Distribution of Sleep Duration')
plt.xlabel('Sleep Duration (hours)')
plt.ylabel('Frequency')

plt.subplot(3, 2, 2)
health['Quality_of_Sleep'].plot.hist(bins=20, color='lightgreen', edgecolor='black')
plt.title('Distribution of Quality of Sleep')
plt.xlabel('Quality of Sleep')
plt.ylabel('Frequency')

plt.subplot(3, 2, 3)
health['Physical_Activity_Level'].plot.hist(bins=20, color='lightcoral', edgecolor='black')
plt.title('Distribution of Physical Activity Level')
plt.xlabel('Physical Activity Level')
plt.ylabel('Frequency')
```

```
plt.subplot(3, 2, 4)
health['Stress_Level'].plot.hist(bins=20, color='lightpink', edgecolor='black')
plt.title('Distribution of Stress Level')
plt.xlabel('Stress Level')
plt.ylabel('Frequency')

plt.subplot(3, 2, 5)
health['Heart_Rate'].plot.hist(bins=20, color='lightgoldenrodyellow', edgecolor='black')
plt.title('Distribution of Heart Rate')
plt.xlabel('Heart Rate (bpm)')
plt.ylabel('Frequency')

plt.subplot(3, 2, 6)
health['Daily_Steps'].plot.hist(bins=20, color='lightskyblue', edgecolor='black')
plt.title('Distribution of Daily Steps')
plt.xlabel('Daily Steps')
plt.ylabel('Frequency')

plt.suptitle('Univariate Analysis: Histograms of Numeric Variables', fontsize=18)
plt.tight_layout(rect=[0, 0, 1, 0.96])
plt.show()
```



Explanation:
plt.figure(figsize=(18, 12))

- This line sets the size of the overall figure (the entire collection of subplots). It specifies the width as 18 inches and the height as 12 inches, which is suitable for displaying multiple histograms clearly.

The code uses plt.subplot() to arrange the histograms in a grid layout of 3 rows and 2 columns.
First Histogram (Sleep Duration)::

- This histogram displays the distribution of the amount of sleep (in hours) reported by individuals. It helps to understand the frequency of different sleep durations and if there is any skewness, such as more people sleeping a specific number of hours.

Second Histogram (Quality of Sleep):

- This histogram shows how the self-reported quality of sleep is distributed. It could represent ratings or a score that reflects how individuals rate their sleep quality (e.g., from "poor" to "excellent"). The histogram visualizes the most common ratings or scores, helping to identify if sleep quality is generally high or low across the dataset.

Third Histogram (Physical Activity Level):

- This histogram visualizes the frequency of different levels of physical activity. The variable might represent activity levels such as sedentary, light, moderate, or intense activity. The histogram will help identify the distribution of activity levels across the dataset.

Fourth Histogram (Stress Level):

- This histogram represents the distribution of reported stress levels. It visualizes how many individuals report different levels of stress, which can help identify the overall stress patterns in the dataset.

Fifth Histogram (Heart Rate):

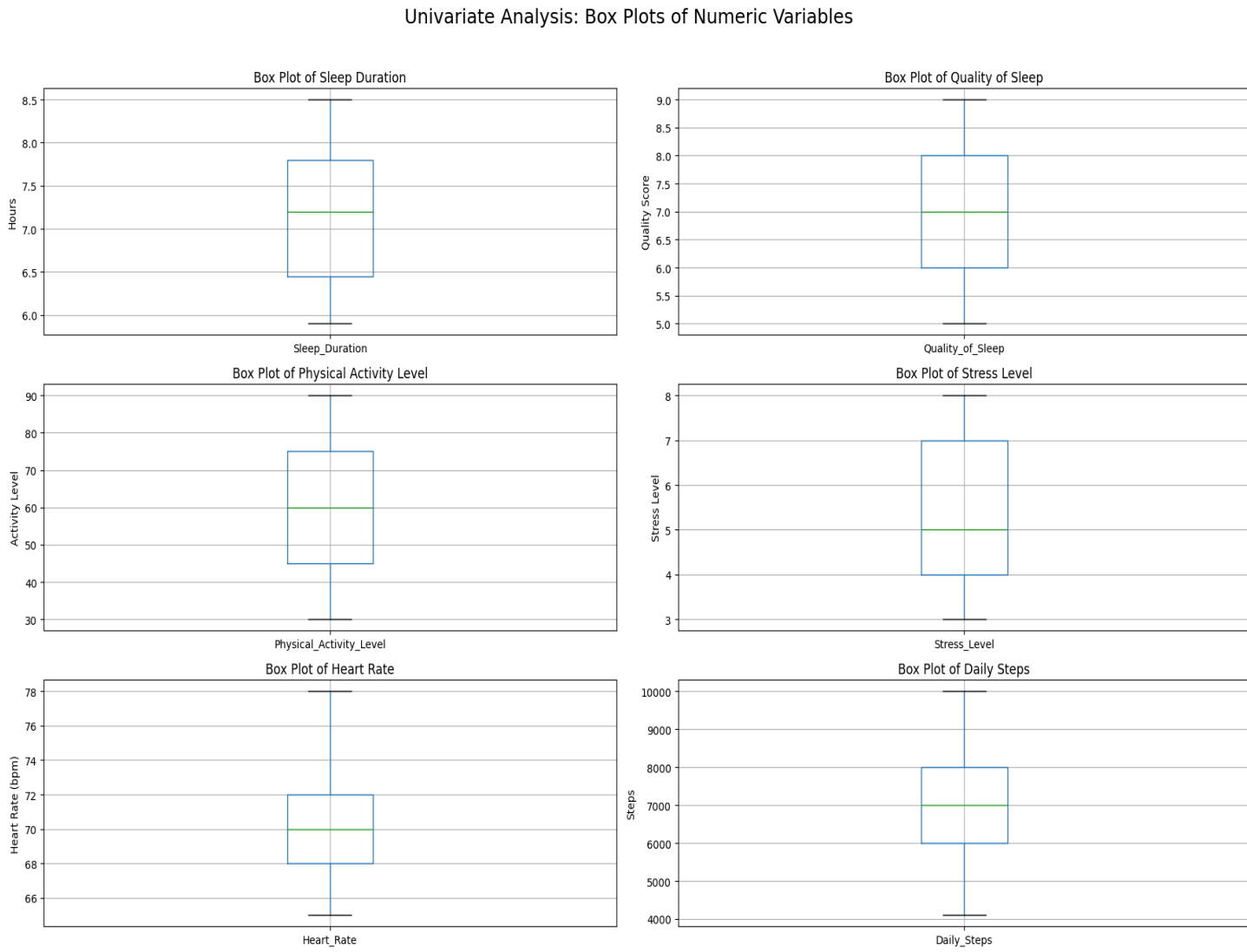
- This histogram shows the distribution of heart rate values (beats per minute). It helps visualize how frequently different heart rate values occur, such as whether most individuals have a normal heart rate, or if there is a variation with higher or lower values.

Sixth Histogram (Daily Steps):

- This histogram visualizes the number of daily steps taken by individuals. It shows the distribution of activity levels in terms of how many steps people take each day. It can help understand the variation in activity levels, such as if most people are leading a sedentary lifestyle or are more active.

PROGRAM:

```
plt.figure(figsize=(18, 12))
plt.subplot(3, 2, 1)
health.boxplot(column='Sleep_Duration')
plt.title('Box Plot of Sleep Duration')
plt.ylabel('Hours')

plt.subplot(3, 2, 2)
health.boxplot(column='Quality_of_Sleep')
plt.title('Box Plot of Quality of Sleep')
plt.ylabel('Quality Score')

plt.subplot(3, 2, 3)
health.boxplot(column='Physical_Activity_Level')
plt.title('Box Plot of Physical Activity Level')
plt.ylabel('Activity Level')

plt.subplot(3, 2, 4)
health.boxplot(column='Stress_Level')
plt.title('Box Plot of Stress Level')
plt.ylabel('Stress Level')

plt.subplot(3, 2, 5)
health.boxplot(column='Heart_Rate')
plt.title('Box Plot of Heart Rate')
plt.ylabel('Heart Rate (bpm)')

plt.subplot(3, 2, 6)
health.boxplot(column='Daily_Steps')
plt.title('Box Plot of Daily Steps')
plt.ylabel('Steps')

plt.suptitle('Univariate Analysis: Box Plots of Numeric Variables', fontsize=18)
plt.tight_layout(rect=[0, 0, 1, 0.96])
plt.show()
```



Explanation:

Box Plot 1: Sleep Duration

The first box plot visualizes the distribution of **Sleep Duration** in hours. The box represents the interquartile range (IQR) where the middle 50% of the data lies, with the median value marked by a line inside the box. The whiskers show the range of the data excluding outliers, which are represented by dots outside the whiskers. This box plot allows us to assess the central tendency, spread, and potential outliers in sleep duration among the subjects.

Box Plot 2: Quality of Sleep

The second box plot shows the distribution of **Quality of Sleep** scores. Similar to the first box plot, it illustrates the IQR and the median. Outliers are also shown to help identify extreme

values. This visualization helps to understand the variability in the perceived quality of sleep among the subjects.

Box Plot 3: Physical Activity Level

The third box plot illustrates the distribution of **Physical Activity Level**. It shows how physical activity levels vary, with the IQR representing the spread of activity levels in the dataset. Outliers are again marked to indicate participants with significantly different activity levels. This plot provides insights into the general physical activity patterns across the dataset.

Box Plot 4: Stress Level

The fourth box plot presents the **Stress Level** distribution. The plot helps visualize the central tendency and spread of stress levels, highlighting any outliers that could represent unusually high or low stress levels among the participants. It provides insights into how stress is distributed across the sample.

Box Plot 5: Heart Rate

The fifth box plot shows the distribution of **Heart Rate (bpm)**. It reveals the central range of heart rate values, the median, and any extreme values. This helps to identify whether heart rates are consistent or if there are unusual spikes or drops in the data that may need further investigation.

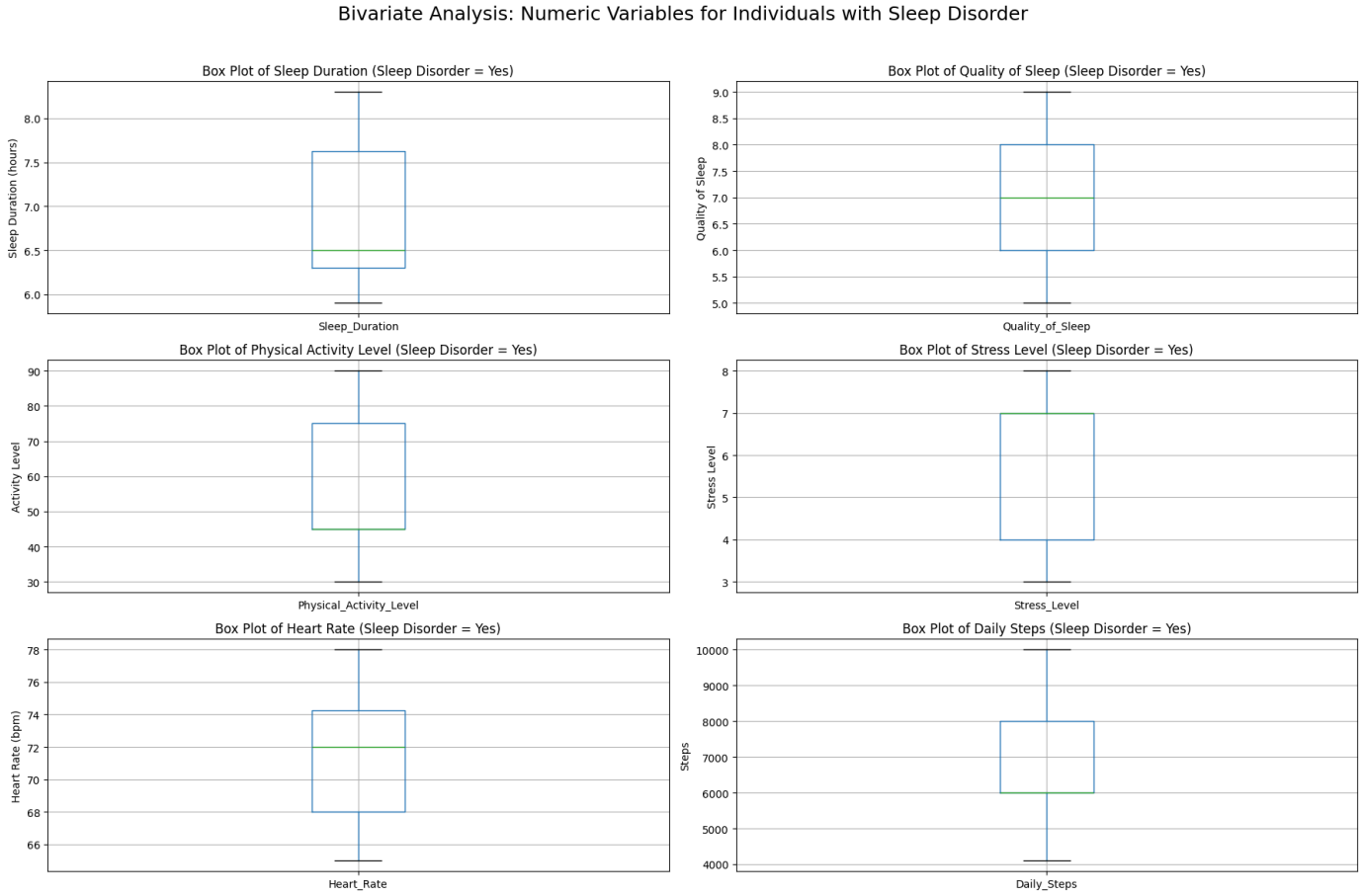
Box Plot 6: Daily Steps

The final box plot illustrates the distribution of **Daily Steps** taken. The central range of daily steps, the median, and potential outliers are visualized, providing insight into how much physical activity participants typically engage in throughout the day. The box plot can also highlight any extreme values, such as participants with very high or low step counts.

PROGRAM:

```
# Filter data to include only rows where Sleep_Disorder is 'yes'
sleep_disorder_data = health[health['Sleep_Disorder'] == 'Yes']

# Define figure size for box plots of numeric variables with Sleep_Disorder = 'yes'
plt.figure(figsize=(18, 12))

# Box plot for Sleep_Duration
plt.subplot(3, 2, 1)
sleep_disorder_data.boxplot(column='Sleep_Duration')
```

```
plt.title('Box Plot of Sleep Duration (Sleep Disorder = Yes)')
plt.ylabel('Sleep Duration (hours)')

# Box plot for Quality_of_Sleep
plt.subplot(3, 2, 2)
sleep_disorder_data.boxplot(column='Quality_of_Sleep')
plt.title('Box Plot of Quality of Sleep (Sleep Disorder = Yes)')
plt.ylabel('Quality of Sleep')

# Box plot for Physical_Activity_Level
plt.subplot(3, 2, 3)
sleep_disorder_data.boxplot(column='Physical_Activity_Level')
plt.title('Box Plot of Physical Activity Level (Sleep Disorder = Yes)')
plt.ylabel('Activity Level')

# Box plot for Stress_Level
plt.subplot(3, 2, 4)
sleep_disorder_data.boxplot(column='Stress_Level')
plt.title('Box Plot of Stress Level (Sleep Disorder = Yes)')
plt.ylabel('Stress Level')

# Box plot for Heart_Rate
plt.subplot(3, 2, 5)
sleep_disorder_data.boxplot(column='Heart_Rate')
plt.title('Box Plot of Heart Rate (Sleep Disorder = Yes)')
plt.ylabel('Heart Rate (bpm)')

# Box plot for Daily_Steps
plt.subplot(3, 2, 6)
sleep_disorder_data.boxplot(column='Daily_Steps')
plt.title('Box Plot of Daily Steps (Sleep Disorder = Yes)')
plt.ylabel('Steps')

plt.suptitle('Bivariate Analysis: Numeric Variables for Individuals with Sleep Disorder',
fontsize=18)
plt.tight_layout(rect=[0, 0, 1, 0.96])
plt.show()
```



Explanation:

Sleep Duration:

The median sleep duration for individuals with sleep disorders appears to be around 7 hours, with a typical range between approximately 6.5 and 8 hours.

There may be some outliers, but overall, the range is relatively narrow. This could suggest that people with sleep disorders tend to get an average amount of sleep but may still experience poor quality, disruptions, or other issues affecting sleep health.

Quality of Sleep:

The median quality of sleep rating for individuals with sleep disorders appears low, around 7 on the scale.

The range is from 5 to 9, indicating that, on average, these individuals may not feel highly rested or satisfied with their sleep. Low-quality sleep is typical for people with sleep disorders, and this distribution reflects that.

Physical Activity Level:

The median physical activity level for individuals with sleep disorders is moderate, with a range from about 30 to 80.

There is considerable variability in activity levels among individuals with sleep disorders, which might suggest that physical activity doesn't necessarily prevent sleep issues for this group.

Stress Level:
The median stress level is around 5, with a range from 3 to 8.
Higher stress levels are common among people with sleep disorders, as indicated by the broad spread. This suggests that stress may be a contributing factor to sleep disorders, though not everyone with sleep disorders has high stress.

Heart Rate:
The median heart rate is around 72 bpm, with a range between 66 and 78 bpm.
Although the range here is narrow, elevated heart rates could indicate physiological stress or poor recovery due to sleep disturbances. Higher heart rates at rest are sometimes associated with stress or poor sleep quality.

Daily Steps:
The median daily steps for individuals with sleep disorders is around 7000, with a range between approximately 4000 and 10,000.
This range indicates variability in activity levels. Some individuals may have low step counts, possibly due to fatigue or reduced energy levels related to poor sleep, while others maintain a relatively active lifestyle.

PROGRAM:

```
# Define the variables and target variable
variables = ['Sleep_Duration', 'Quality_of_Sleep', 'Physical_Activity_Level', 'Stress_Level',
'Heart_Rate', 'Daily_Steps']
target_var = 'Sleep_Disorder'

plt.figure(figsize=(18, 12))

# Loop through each variable to create subplots
for i, var in enumerate(variables, 1):
    # Create crosstab and normalize by row
    M = pd.crosstab(health[var], health[target_var], normalize='index')
    print(f'Crosstab for {var} vs {target_var}:\n', M)

    # Plot in a 3x2 grid layout
    plt.subplot(3, 2, i)
    M.plot.bar(stacked=False, ax=plt.gca())  # Using ax=plt.gca() to plot within each subplot
    plt.title(f'{var} vs {target_var}')
```

```
    plt.xlabel(var)
    plt.ylabel('Proportion')
    plt.legend(title=f'{target_var}')

# Add a main title and adjust layout
plt.suptitle('Bivariate Analysis: Crosstabs of Variables with Target', fontsize=18)
plt.tight_layout(rect=[0, 0, 1, 0.96])
```

Crosstab for Sleep_Duration vs Sleep_Disorder:

Sleep_Disorder	No	Yes
Sleep_Duration		
5.9	0.000000	1.000000
6.0	0.548387	0.451613
6.1	0.440000	0.560000
6.2	0.750000	0.250000
6.3	0.000000	1.000000
6.4	0.111111	0.888889
6.5	0.043478	0.956522
6.6	0.100000	0.900000
6.7	0.400000	0.600000
6.8	0.000000	1.000000
6.9	1.000000	0.000000
7.1	0.947368	0.052632
7.2	0.916667	0.083333
7.3	1.000000	0.000000
7.4	1.000000	0.000000
7.5	1.000000	0.000000
7.6	0.900000	0.100000
7.7	0.958333	0.041667
7.8	0.892857	0.107143
7.9	1.000000	0.000000
8.0	0.230769	0.769231
8.1	0.153846	0.846154
8.2	0.181818	0.818182
8.3	0.800000	0.200000
8.4	1.000000	0.000000
8.5	1.000000	0.000000

Crosstab for Quality_of_Sleep vs Sleep_Disorder:

Sleep_Disorder	No	Yes
Quality_of_Sleep		

5 0.000000 1.000000
6 0.384615 0.615385
7 0.547945 0.452055
8 0.926606 0.073394
9 0.550725 0.449275

Crosstab for Physical_Activity_Level vs Sleep_Disorder:

Sleep_Disorder	No	Yes
Physical_Activity_Level		
30	0.923077	0.076923
35	0.000000	1.000000
40	1.000000	0.000000
42	1.000000	0.000000
45	0.088235	0.911765
47	1.000000	0.000000
50	1.000000	0.000000
55	1.000000	0.000000
60	0.941176	0.058824
70	1.000000	0.000000
75	0.537313	0.462687
80	1.000000	0.000000
90	0.492537	0.507463

Crosstab for Stress_Level vs Sleep_Disorder:

Sleep_Disorder	No	Yes
Stress_Level		
3	0.579710	0.420290
4	0.614286	0.385714
5	0.904762	0.095238
6	0.934783	0.065217
7	0.065217	0.934783
8	0.507692	0.492308

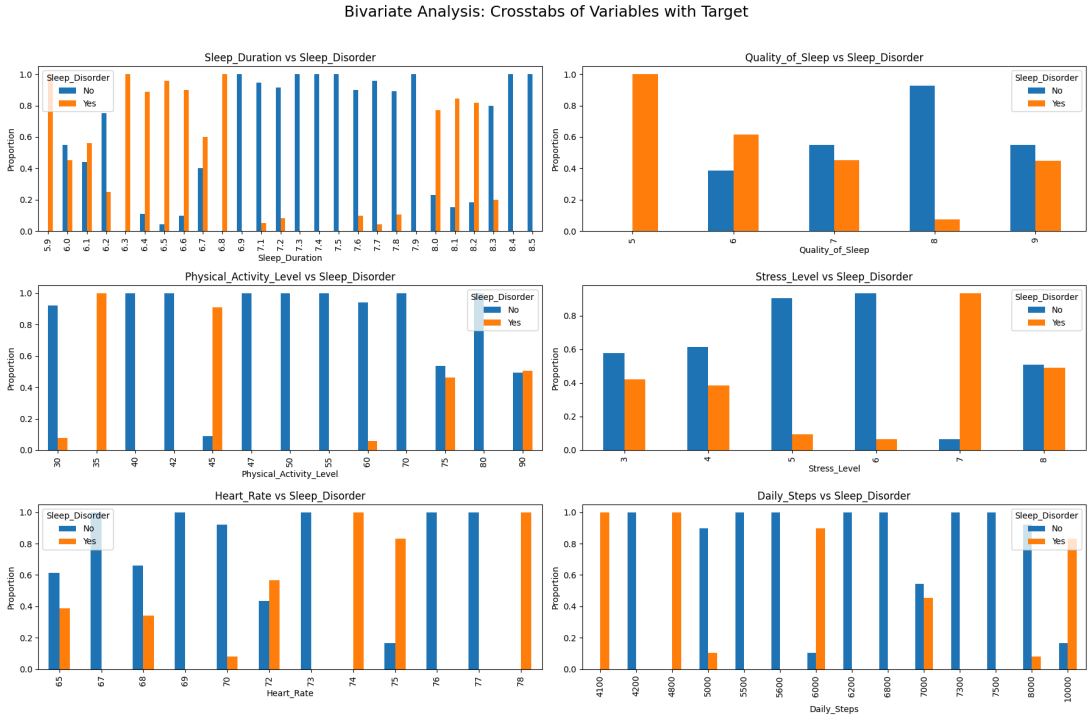
Crosstab for Heart_Rate vs Sleep_Disorder:

Sleep_Disorder	No	Yes
Heart_Rate		
65	0.611940	0.388060
67	1.000000	0.000000
68	0.659574	0.340426
69	1.000000	0.000000
70	0.921053	0.078947
72	0.434783	0.565217
73	1.000000	0.000000

74	0.000000	1.000000
75	0.166667	0.833333
76	1.000000	0.000000
77	1.000000	0.000000
78	0.000000	1.000000

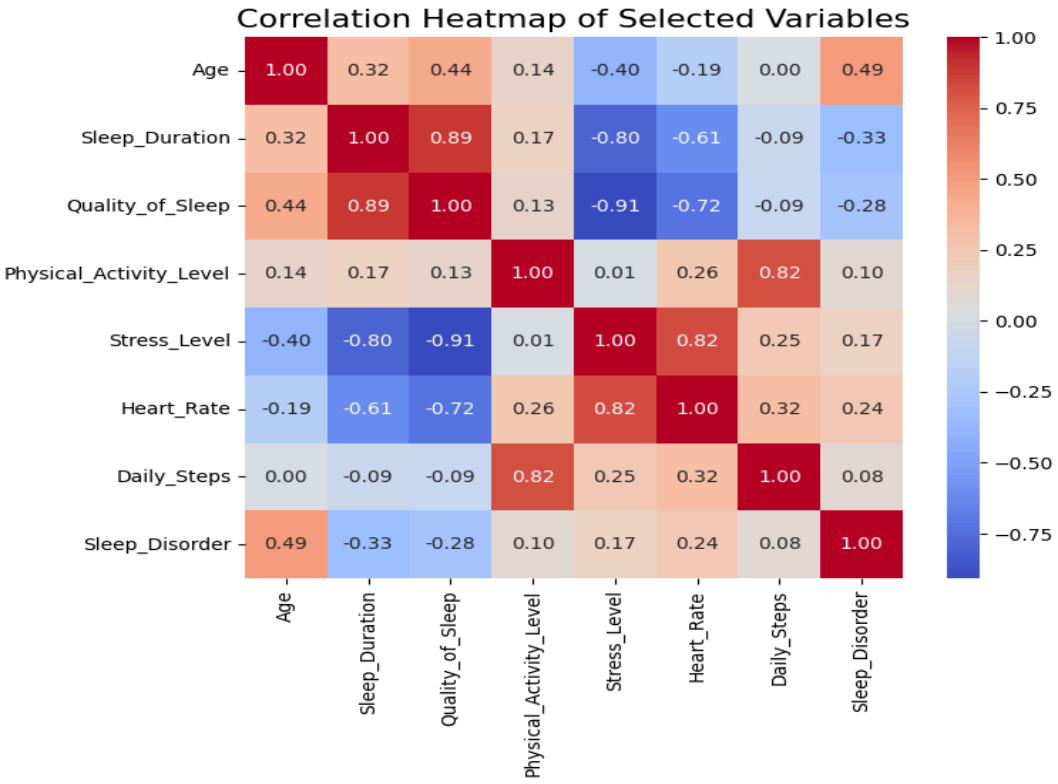
Crosstab for Daily_Steps vs Sleep_Disorder:

Sleep_Disorder	No	Yes
Daily_Steps		
4100	0.000000	1.000000
4200	1.000000	0.000000
4800	0.000000	1.000000
5000	0.897059	0.102941
5500	1.000000	0.000000
5600	1.000000	0.000000
6000	0.102941	0.897059
6200	1.000000	0.000000
6800	1.000000	0.000000
7000	0.545455	0.454545
7300	1.000000	0.000000
7500	1.000000	0.000000
8000	0.920792	0.079208
10000	0.166667	0.833333



PROGRAM:

```
# Select the subset of data with the specified variables
var = health[['Age','Sleep_Duration', 'Quality_of_Sleep', 'Physical_Activity_Level',
                    'Stress_Level', 'Heart_Rate', 'Daily_Steps','Sleep_Disorder']]

# Calculate the correlation matrix for these selected variables
corr_matrix = var.corr()

# Set up the matplotlib figure
plt.figure(figsize=(8, 6))

# Generate a heatmap for the correlation matrix
sns.heatmap(corr_matrix, annot=True, fmt=".2f", cmap="coolwarm", cbar=True, square=True)

# Add a title
plt.title('Correlation Heatmap of Selected Variables', fontsize=16)

# Show the plot
plt.show()
```



4. Methodology:

Linear Regression for Classification of Sleep Disorder

Linear regression analysis is used to predict the value of a variable based on the value of another variable. The variable you want to predict is called the dependent variable. The variable you are using to predict the other variable's value is called the independent variable.

PROGRAM:

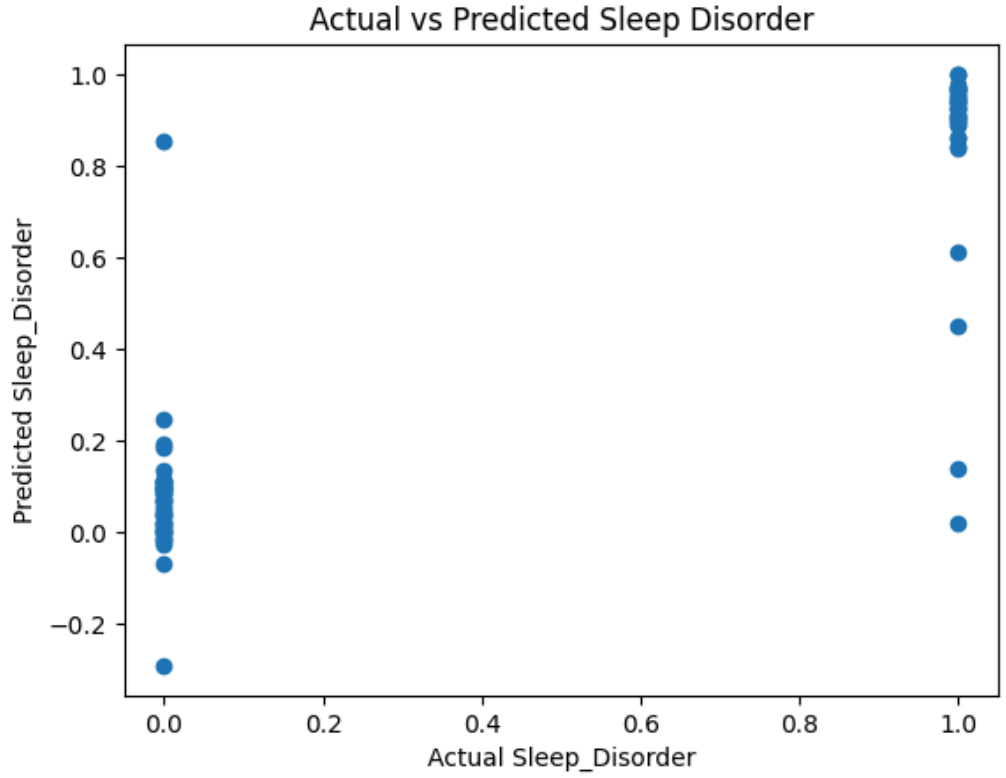
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import numpy as np
health =
pd.read_csv("C:\\Users\\vaish\\Downloads\\Sleep_health_and_lifestyle_dataset_new.csv")
# First, encode the target variable 'Sleep_Disorder' (Yes = 1, No = 0)
health['Sleep_Disorder'] = health['Sleep_Disorder'].map({'Yes': 1, 'No': 0})

health = pd.get_dummies(health) # convert categorical variables into one-hot encoded variables.

# Define features (X) and target variable (y)
X = health.drop('Sleep_Disorder', axis=1)
y = health['Sleep_Disorder']
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create a linear regression model
model = LinearRegression()

# Fit the model on the training data
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

rmse = np.sqrt(mean_squared_error(y_test, y_pred))
print(f'Root Mean Squared Error:{rmse}')
```

```
Root Mean Squared Error:0.21551895197182608
```

PROGRAM:

```
import matplotlib.pyplot as plt
plt.scatter(y_test, y_pred)
plt.xlabel('Actual Sleep_Disorder')
plt.ylabel('Predicted Sleep_Disorder')
plt.title('Actual vs Predicted Sleep Disorder')
```

```
Text(0.5, 1.0, 'Actual vs Predicted Sleep Disorder')
```

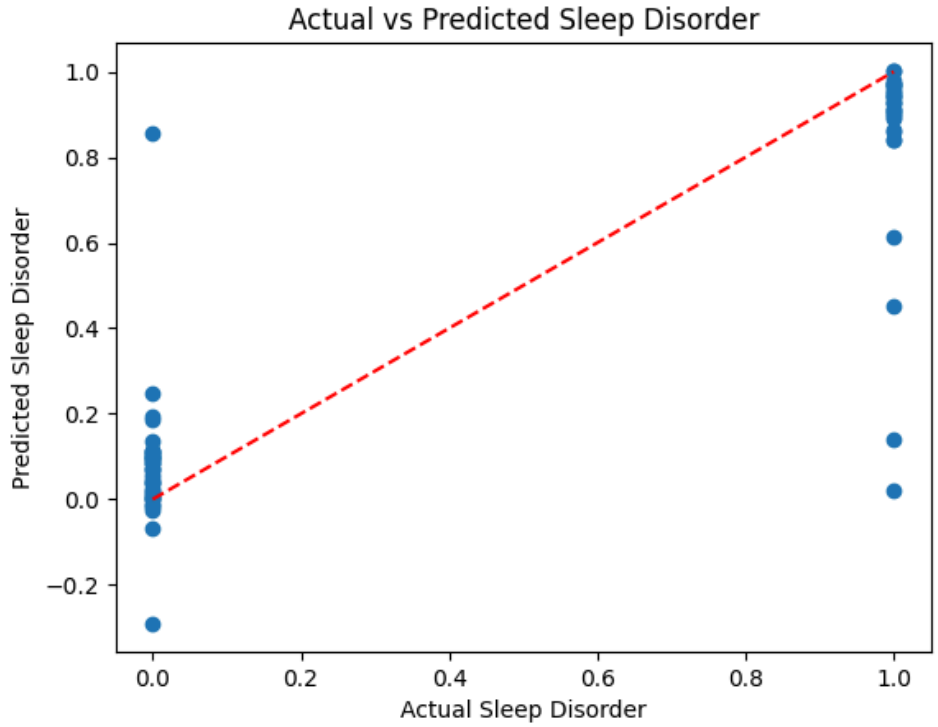


Each point in the plot represents a single test sample. The **x-axis** shows the **actual value** of Sleep_Disorder(0 or 1), and the **y-axis** shows the **predicted value** (also between 0 and 1).
The goal is to see if the predictions are close to the actual values. A good model will have points near a 45-degree line where y_test = y_pred.

PROGRAM:

```
plt.scatter(y_test, y_pred)
plt.xlabel('Actual Sleep Disorder')
plt.ylabel('Predicted Sleep Disorder')
```

```
plt.title('Actual vs Predicted Sleep Disorder')
plt.plot([y.min(), y.max()], [y.min(), y.max()], 'r--')  # line of equality
plt.show()
```



The line [y.min(), y.max()], [y.min(), y.max()] creates a **line of equality**—a diagonal line where y_test = y_pred. This is the line that represents perfect predictions. If the model's predictions were perfect, all points would lie on this line.

- The line is plotted using plt.plot(), with 'r--' specifying that it should be a red dashed line.

Interpretation:

- **Points close to the line**: The model's predictions are accurate.
- **Points far from the line**: The model's predictions are not accurate and need improvement.

PROGRAM:

```
# Model Evaluation
# Make predictions on the test set
y_pred = model.predict(X_test)

# Calculate performance metrics
rmse = np.sqrt(mean_squared_error(y_test, y_pred))  # Calculate RMSE
```

```
r2 = r2_score(y_test, y_pred)

# Print performance metrics and model coefficients
print("Root Mean Squared Error (RMSE):", rmse)
print("R-squared:", r2)

# Display the model coefficients
print("Coefficients:", model.coef_)
print("Intercept:", model.intercept_)
```

```
Root Mean Squared Error (RMSE): 0.21551895197182608
R-squared: 0.8101218350602706
Coefficients: [-2.67239690e-03  3.21801284e-02 -2.66535001e-01  2.37613861e-01
  2.77281101e-03  6.32248248e-02  1.26046573e-02 -4.95880734e-05
  1.78038453e-01 -1.78038453e-01 -5.21047066e-02  1.25884600e-01
  7.23292511e-02 -6.72113013e-02 -4.03954672e-01 -2.22989818e-01
  4.27090672e-01  4.84713369e-01 -3.74222685e-01 -1.19281049e-01
  1.29746340e-01 -2.20901504e-01 -1.91048752e-01  2.76927706e-01
  1.35022551e-01 -3.35985047e-01 -3.14482795e-01  1.56318924e-02
 -3.51675692e-01 -7.57931096e-02 -4.96108059e-01  9.34715532e-02
 -3.41500341e-01 -3.57663251e-01 -2.06940407e-01 -2.53571415e-01
 -6.89127787e-01 -5.34158419e-01 -4.27849125e-01  7.35681326e-01
  1.58918854e-01  9.17970396e-01  7.86214470e-01  8.15457240e-01
  0.00000000e+00  2.38675778e-01 -1.91225483e-03  6.04020035e-02
  3.45906234e-01  2.18437957e-01]
Intercept: -1.363577053879287
```

Model Training and Evaluation (Regression)

A regression model was evaluated using performance metrics:

- **Predictions on Test Set**: The model was used to make predictions on the test data (`y_pred = model.predict(X_test)`).
- **Root Mean Squared Error (RMSE)**: RMSE was calculated to measure the model's prediction error. A lower RMSE indicates better model performance. Here, the RMSE value of 0.2155 suggests that the model's predictions are relatively close to the actual values.
- **R-squared (R^2)**: R-squared measures the proportion of variance in the target variable explained by the predictors. An R^2 value of 0.8101 indicates that about 81% of the variance in the target variable is explained by the model, which demonstrates a good fit.

- **Model Coefficients and Intercept**: The model's coefficients and intercept were displayed to show the relationship between each predictor and the target variable. The coefficients indicate the degree and direction of influence each feature has on the target. For example, positive coefficients indicate a positive relationship, while negative coefficients indicate an inverse relationship.

Logistic Regression for Classification of Sleep Disorder

PROGRAM:

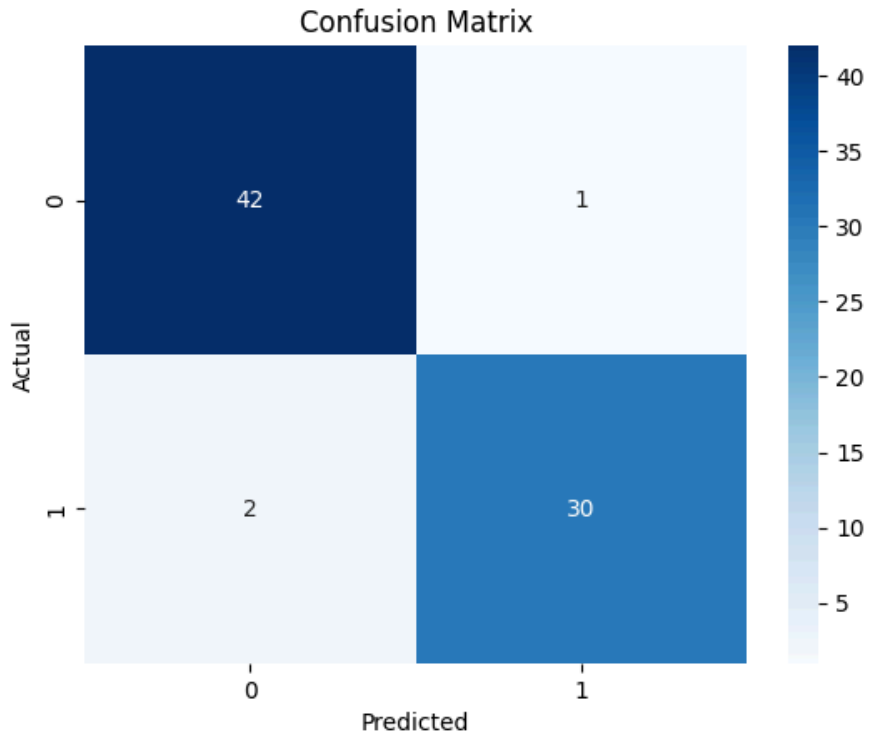
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix
from sklearn.preprocessing import StandardScaler

health_encoded = pd.get_dummies(health, drop_first=True)  # Drop the first category to avoid the dummy variable trap

# Define the predictors (X) and target variable (y)
X = health_encoded.drop(columns=['Sleep_Disorder'])  # Drop the target column from predictors
y = health_encoded['Sleep_Disorder']  # Target variable

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Feature scaling using StandardScaler
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Train the Logistic Regression model
model = LogisticRegression()
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')
```

```
# Display classification report
print("Classification Report:\n", classification_report(y_test, y_pred))
```

OUTPUT:

```
Accuracy: 0.96
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.98      0.97        43
           1       0.97      0.94      0.95        32

    accuracy                           0.96        75
   macro avg       0.96      0.96      0.96        75
weighted avg       0.96      0.96      0.96        75
```

```
# Display confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)
sns.heatmap(conf_matrix, annot=True, cmap='Blues', fmt='g')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```



Explanation of the Confusion Matrix:

The confusion matrix shown in the image provides a summary of the model's performance in classifying sleep disorders.

True Positives (Bottom Right, 30): The model correctly identified 30 cases where individuals actually have a sleep disorder (Actual = 1) and the model also predicted a sleep disorder (Predicted = 1).

1. **True Negatives (Top Left, 42)**: The model correctly identified 42 cases where individuals do not have a sleep disorder (Actual = 0) and the model also predicted no sleep disorder (Predicted = 0).
2. **False Positives (Top Right, 1)**: The model incorrectly predicted 1 individual as having a sleep disorder (Predicted = 1) when they actually do not have one (Actual = 0). This is also called a **Type I error**.
3. **False Negatives (Bottom Left, 2)**: The model incorrectly predicted 2 individuals as not having a sleep disorder (Predicted = 0) when they actually do have one (Actual = 1). This is also called a **Type II error**.

Key Metrics

Based on the confusion matrix, we can calculate a few important metrics:

- **Accuracy**: The overall percentage of correct predictions. Here, it would be 42+30/42+1+2+30, which is quite high, indicating good model performance.
- **Precision for Class 1 (Sleep Disorder)**: 30/30+1, which measures the accuracy of positive predictions.
- **Recall for Class 1 (Sleep Disorder)**:30/30+2, which measures how well the model identifies all actual positive cases.

This confusion matrix indicates that the model performs well, with a high accuracy and relatively low error rates. The false negative rate is slightly higher than the false positive rate, suggesting that in a few cases, individuals with a sleep disorder might be missed.

Logistic Regression for Classification of Sleep Disorder

In the second part, a Logistic Regression model was used to predict whether an individual has a sleep disorder (Sleep_Disorder). Logistic Regression was chosen as it is well-suited for binary classification problems.

Steps in Preprocessing and Model Evaluation:

- **Encoding Categorical Variables**: pd.get_dummies was used to convert categorical variables into numerical format. This process helps the model handle categorical data by creating binary variables (dummy variables) for each category.
- **Defining Predictors and Target**: The predictors (X) included all columns except for Sleep_Disorder, which is the target variable (y). This ensures that the model only learns from independent features.
- **Splitting the Data**: The dataset was split into training and test sets, with 20% of the data allocated for testing. This is a common approach in machine learning to evaluate model performance on unseen data.
- **Feature Scaling**: StandardScaler was used to standardize the features. Scaling helps improve the model's performance and ensures that features with larger values do not dominate the model's learning.
- **Model Training**: The Logistic Regression model was trained on the scaled training data, allowing it to learn the relationship between the predictors and the target variable.

Model Evaluation:

- **Accuracy Score**: The accuracy score of 0.96 indicates that the model correctly classified 96% of the instances in the test set. This high accuracy reflects the model's effectiveness in distinguishing between individuals with and without a sleep disorder.
- **Classification Report**:
    - The classification report provides detailed metrics for each class (0 = no sleep disorder, 1 = sleep disorder):

- **Precision**: The proportion of true positive predictions to the total predicted positives. A high precision score (0.95 for class 0, 0.97 for class 1) suggests that the model accurately identifies individuals without a sleep disorder.
- **Recall**: The proportion of true positives to the total actual positives. The recall of 0.98 for class 0 and 0.94 for class 1 shows that the model is good at identifying individuals with a sleep disorder.
- **F1-Score**: The F1-score is the harmonic mean of precision and recall. The F1-scores of 0.97 (for class 0) and 0.95 (for class 1) indicate a balance between precision and recall.

PROGRAM:

```
# Convert categorical variables to numeric

label_encoders = {}
for column in ['Gender', 'Occupation', 'BMI_Category']:
    le = LabelEncoder()
    health[column] = le.fit_transform(health[column])
    label_encoders[column] = le

# Split Blood Pressure into two columns

health[['Systolic_BP', 'Diastolic_BP']] = health['Blood_Pressure'].str.split('/',
expand=True).astype(float)
health.drop(columns=['Blood_Pressure', 'Person_ID'], inplace=True)

# Encode target variable
health['Sleep_Disorder'] = health['Sleep_Disorder'].map({'No': 0, 'Yes': 1})

# Separate features and target
X = health.drop(columns=['Sleep_Disorder'])
y = health['Sleep_Disorder']

# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Scale features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
# Train a Random Forest Classifier
clf = RandomForestClassifier(random_state=42)
clf.fit(X_train, y_train)

# Evaluate model on test set
y_pred = clf.predict(X_test)
print("Model performance on test data:")
print(classification_report(y_test, y_pred))

# Step 3: Predict future risk for those without current sleep disorders
# Filter data for people currently without a sleep disorder
no_disorder = health[health['Sleep_Disorder'] == 0].drop(columns=['Sleep_Disorder'])

# Scale the features for these individuals
no_disorder_scaled = scaler.transform(no_disorder)

# Predict future risk
future_risk_predictions = clf.predict(no_disorder_scaled)

# Add predictions to the original data
health.loc[health['Sleep_Disorder'] == 0, 'Future_Disorder_Risk'] = future_risk_predictions

# Count of people predicted to potentially develop a sleep disorder
at_risk_count = (future_risk_predictions == 1).sum()
print(f"Number of people without a current sleep disorder predicted to develop one in the future:
{at_risk_count}")

### Step 4: Feature Importance
# Get feature importance from the trained model
feature_importances = clf.feature_importances_

# Create a DataFrame to easily view and sort the feature importances
importance_df = pd.DataFrame({'Feature': X.columns, 'Importance': feature_importances})
importance_df = importance_df.sort_values(by='Importance', ascending=False)

print("\nFeature Importances:")
print(importance_df)

# Plot the feature importances for visualization
```

```
plt.figure(figsize=(10, 6))
plt.barh(importance_df['Feature'], importance_df['Importance'], color='skyblue')
plt.gca().invert_yaxis()
plt.xlabel('Importance')
plt.ylabel('Feature')
plt.title('Feature Importance for Sleep Disorder Prediction')
plt.show()
```

OUTPUT:

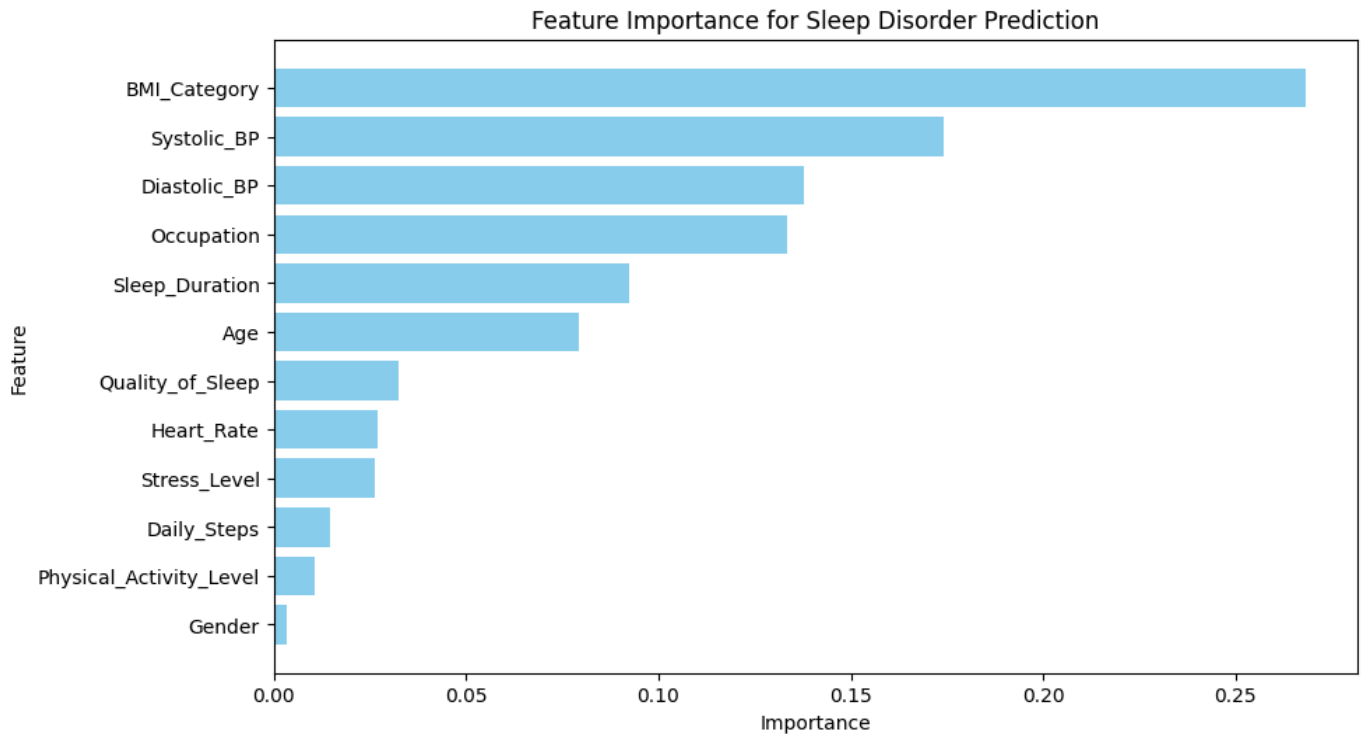
Model performance on test data:

	precision	recall	f1-score	support
0	0.95	0.98	0.97	43
1	0.97	0.94	0.95	32
accuracy			0.96	75
macro avg	0.96	0.96	0.96	75
weighted avg	0.96	0.96	0.96	75

Number of people without a current sleep disorder predicted to develop one in the future: 9

Feature Importances:

	Feature	Importance
7	BMI_Category	0.268062
10	Systolic_BP	0.174193
11	Diastolic_BP	0.137797
2	Occupation	0.133480
3	Sleep_Duration	0.092508
1	Age	0.079186
4	Quality_of_Sleep	0.032396
8	Heart_Rate	0.026901
6	Stress_Level	0.026476
9	Daily_Steps	0.014802
5	Physical_Activity_Level	0.010630
0	Gender	0.003569



Feature Importance Analysis for Sleep Disorder Prediction

The chart above illustrates the feature importance scores derived from the Random Forest model used to predict the likelihood of developing a sleep disorder. Feature importance reflects the relative contribution of each factor in the model's decision-making process, helping to identify which factors are most influential.

1. **Top Contributing Features**:
    - **BMI Category**: The Body Mass Index (BMI) category stands out as the most critical predictor, with the highest importance score (over 0.25). This suggests that BMI, which reflects overall body weight in relation to height, plays a significant role in determining the risk of developing a sleep disorder.
    - **Systolic and Diastolic Blood Pressure**: Both systolic and diastolic blood pressure measurements have high importance scores. These metrics indicate the effect of cardiovascular health on sleep disorders, suggesting that higher or fluctuating blood pressure could be linked to increased risk.
    - **Occupation**: The type of occupation also plays a significant role, likely due to varying levels of stress, physical activity, and work schedules associated with different jobs.
2. **Moderately Contributing Features**:

    - **Sleep Duration**: The amount of sleep an individual gets shows a moderate level of importance, underscoring the relationship between sleep quantity and the risk of sleep disorders.
    - **Age**: Age is also a contributing factor, suggesting that as individuals grow older, their risk of developing a sleep disorder may change.
    - **Quality of Sleep**: Although it has a lower importance score compared to BMI and blood pressure, the quality of sleep remains a relevant factor. This highlights that not only the duration but also the perceived quality of sleep affects sleep health.
3. **Lower Contributing Features**:
    - **Stress Level, Daily Steps, and Physical Activity Level**: These factors have lower importance but are still relevant to the model. While they are less predictive individually, they may have a cumulative or indirect effect on sleep health.
    - **Gender**: Gender shows the lowest importance score, indicating it may have a limited direct impact on predicting sleep disorders in this dataset.

Interpretation:

The feature importance analysis reveals that physiological factors like BMI and blood pressure are the strongest predictors of sleep disorders in this model. This implies a strong link between physical health and sleep health. Lifestyle factors, such as occupation and sleep duration, also contribute significantly, highlighting the role of work-related stress and sleep habits.

5. Conclusion

This study highlights the impact of lifestyle and physiological factors on sleep health, focusing on both current and future sleep disorder risk. Key findings show that high BMI, elevated stress, low physical activity, and poor cardiovascular health are associated with a higher risk of sleep disorders and lower sleep quality. Using a Random Forest model, we also identified individuals who currently do not have sleep disorders but are at risk in the future, with BMI, blood pressure, occupation, and sleep duration emerging as the strongest predictors.

To reduce sleep disorder risk, interventions should focus on promoting physical activity, managing stress, and maintaining healthy weight and blood pressure. These insights can guide targeted strategies to improve sleep quality and overall well-being.

6. References

1. Datasets:

- Sleep Health and Lifestyle Dataset (used in our code):

- Source: www.kaggle.com

2. Machine Learning Algorithms and Techniques:

- Random Forest Classifier:

  "Breiman, L. (2001). Random Forests." *Machine Learning*, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324

- Label Encoding:

  Scikit-learn documentation on LabelEncoder: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

- Feature Importance:

  "Scikit-learn documentation on Random Forest: Feature importance": https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

- StandardScaler:

  Scikit-learn documentation on StandardScaler: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

3. Papers and Research:

- Sleep Disorders and Machine Learning:
  - Gao, K., Chen, X., & Li, L. (2021). Predicting sleep disorder based on machine learning techniques. *Journal of Healthcare Engineering*, 2021, 1-9. https://doi.org/10.1155/2021/6994941
- Health Prediction Using Machine Learning:
  - Buda, M., & Seki, Y. (2018). Machine learning techniques for healthcare data analysis: A survey. *Journal of Healthcare Engineering*, 2018, 1-12. https://doi.org/10.1155/2018/7823536

4. Visualization:

- Matplotlib Documentation:
  - "Matplotlib: Plotting with Python." https://matplotlib.org/stable/contents.html
- Visualization of Feature Importance:

- "Visualizing Feature Importance in Random Forests." *Data Science Handbook* (online resource): https://www.datacamp.com/community/tutorials/random-forests-classifier-python

5. General Resources:

- Scikit-learn Documentation:

  Official documentation for machine learning in Python: https://scikit-learn.org/stable/

- Pandas Documentation:

  Official pandas documentation for data manipulation: https://pandas.pydata.org/pandas-docs/stable/

- YouTube - "Random Forest Algorithm - Simply Explained" by StatQuest with Josh Starmer - https://www.youtube.com/watch?v=J4Wdy0Wc_xQ
- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html