LIFE STYLE CHANGE DUE TO COVID PREDICTION

AN INDUSTRY ORIENTED MINI REPORT

Submitted to

JAWAHARLAL NEHRU TECNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Submitted By

BALABHADRA SAI KIRAN	**21UK1A05G6**
BORIGAM SHARVANI	**21UK1A05H2**
PRASHANTH DUBARLA	**21UK1A05J0**
PUJARI KARTHIK	**21UK1A05K1**

Under the guidance of

P. Vamshi krishna

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

Affiliated to JNTUH, HYDERABAD

BOLLIKUNTA, WARANGAL (T.S) –506005

DEPARTMENT OF

COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE(WARANGAL)

CERTIFICATE OF COMPLETION

INDUSTRY ORIENTED MINI PROJECT

This is to certify that the UG Project Phase-1 entitled “LIFE STYLE CHANGE DUE TO COVID PREDICTION” is being submitted by BALABHADRA SAI KIRAN(21UK1A05G6),BORIGAM SHARVANI(21UK1A05H2),PRASHANTH DUBARLA(21UK1A05J0),

PUJARI KARTHIK(21UK1A05K1) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2024-2025.

Project Guide **HOD**

P. Vamshi krishna **Dr. R.Naveen kumar**

(Professor)

External

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr.P.PRASAD RAO**, Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase-1 in the institute.

We extend our heartfelt thanks to **Dr.R.Naveen kumar** Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase-1.

We express heartfelt thanks to Smartinternz Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the UG Project Phase-1 and for their support in completing the UG Project Phase-1.

We express heartfelt thanks to the guide**, P.Vamshi krishna** , faculty of CSE Department for his constant support and giving necessary guidance for completion of this UG Project Phase-1.

Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

BALABHADRA SAI KIRAN (21UK1A05G6)

BORIGAM SHARVANI (21UK1A05H2)

PRASHANTH DUBARLA (21UK1A05J0)

PUJARI KARTHIK (21UK1A05K1)

ABSTRACT

The COVID-19 pandemic has served as a powerful catalyst for unprecedented lifestyle changes. This prediction explores the potential long-term impact of the virus on daily routines, social interactions, and personal priorities. The abstract examines shifts towards remote work and education, the rise of home-centric activities, and a heightened focus on hygiene. It delves into the evolving nature of social interactions, with virtual gatherings becoming the norm and social distancing impacting relationships. The abstract highlights a potential redefinition of priorities, with health concerns taking center stage alongside a focus on financial security and a newfound appreciation for simplicity. The role of technology in adapting to these changes is explored, with the rise of telehealth, e-commerce, and virtual reality being potential trends. Finally, the abstract acknowledges the uncertainties surrounding the long-term effects of COVID-19, including the impact of new variants and the need to build resilience in a constantly evolving world.

TABLE OF CONTENTS:-

1.INTRODUCTION

1.1.OVERVIEW

The COVID-19 pandemic has been a force for dramatic change, pushing us to re-evaluate how we live on a fundamental level. This overview explores the potential long-term impact of the virus on our daily routines, social interactions, and core values.

Reshaping Daily Life: A shift towards remote work and education may become permanent, altering commutes and work-life balance.

Redefining Social Connections:Virtual gatherings may become the preferred method for socializing, with technology facilitating communication across distances.

Shifting Priorities: Health is likely to remain a top priority, with individuals potentially taking a more proactive approach to preventative care and wellness.

Technological Advancements: Telehealth services are likely to see continued growth, offering convenient access to medical consultations.

Uncertainties and the Path Forward: The long-term societal impact of the pandemic remains to be seen, with potential changes to social structures and economic landscapes.

Building a more resilient society that can navigate change and uncertainty will be crucial in the post-COVID world.

This overview provides a glimpse into the potential future shaped by COVID-19. While the exact nature of these changes remains to be seen, it's clear that the pandemic has ushered in a new era, demanding a reevaluation of how we live, interact, and prioritize our well-being.

1.2.PURPOSE

The main purpose for lifestyle changes due to COVID prediction is two-fold:

Reduce the risk of infection and disease spread: Practices like increased hygiene, social distancing, and potentially even remote work can help mitigate the spread of COVID-19 and similar illnesses.

Adapt to the "new normal" created by the pandemic: Changes in work culture, healthcare access, and social interaction require adjustments in how we live our daily lives to maintain a sense of normalcy and well-being. These adaptations may even lead to positive changes, like a renewed focus on health or a better work-life balance.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

Social isolation and loneliness: Increased reliance on remote work and social distancing can lead to feelings of isolation and a decline in mental well-being.

Work-life balance issues: Blurred lines between work and home life can be a challenge in remote work settings.

Increased screen time and sedentary behavior: Spending more time at home can lead to less physical activity and increased dependence on technology.

Economic hardship: The pandemic's economic impact may force people to adopt less desirable lifestyles, like smaller living spaces or limited access to healthy food.

Uncertainty and Adaptability: Fear and anxiety surrounding the virus: The unpredictable nature of the pandemic can lead to heightened anxiety and difficulty adjusting to new situations.

Information overload and misinformation: Difficulty discerning reliable information can lead to confusion and mistrust.

The potential for "pandemic fatigue: " As the pandemic drags on, people may become less compliant with safety measures, leading to potential resurgence of the virus.

The unknown long-term effects of COVID-19: The long-term health and social impacts of the virus remain unclear, making it difficult to plan for the future.

These existing problems highlight the complexities of adapting to a new normal. While some changes may bring benefits, others pose significant challenges that need to be addressed to ensure a healthy and sustainable future.

2.2 PROPOSED SOLUTION

Here are some proposed solutions for lifestyle changes due to COVID prediction:

Combating Social Isolation and Promoting Well-being: Prioritize virtual social interaction: Encourage online gatherings, virtual game nights, and remote co-working spaces to foster connection.

Develop strong social support networks: Promote community outreach programs and mental health resources to combat feelings of loneliness.

Maintain healthy routines: Encourage regular exercise, hobbies, and activities that contribute to mental and physical well-being.

Enhancing Work-Life Balance in Remote Settings: Establish clear boundaries: Set dedicated work hours, utilize separate workspaces, and avoid checking work emails outside of designated times.

Promote flexible work arrangements: Allow for flexible schedules and breaks to prevent burnout and encourage a healthier work-life balance.

Encourage physical activity breaks: Promote short exercise routines or walking breaks throughout the workday.

Mitigating Screen Time and Sedentary Behavior: Schedule screen-free time: Encourage taking breaks from technology and engaging in outdoor activities or hobbies.

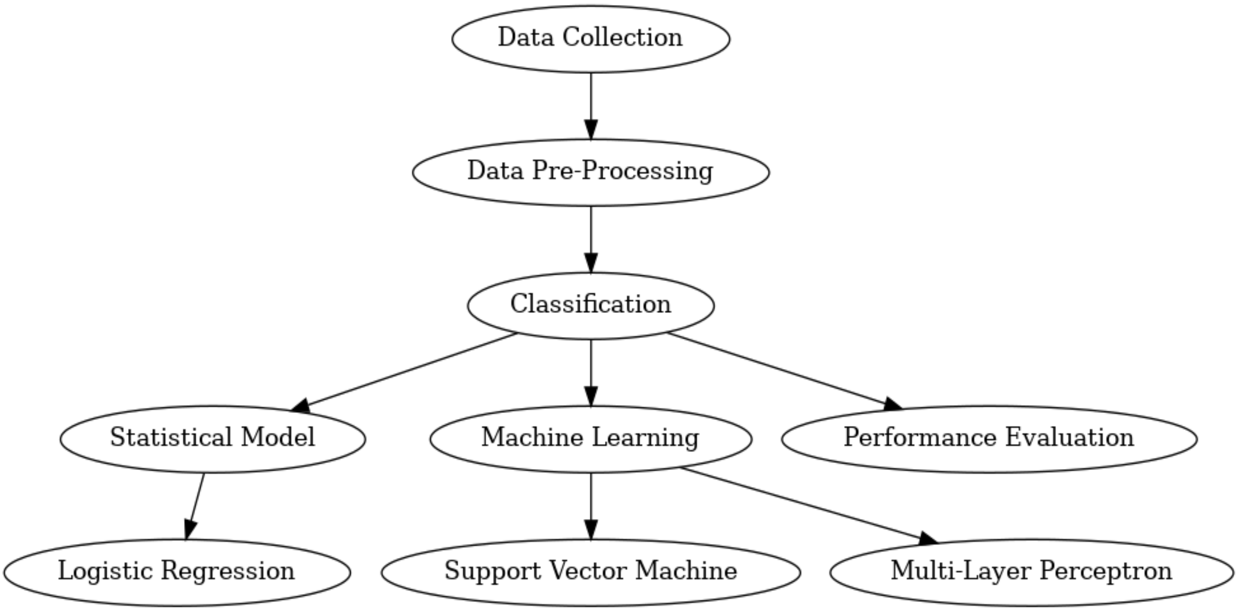
Promote community technology centers: Offer public spaces with internet access and support for those with limited resources.

Develop coping mechanisms for stress and anxiety: Encourage mindfulness practices, stress management techniques, and access to mental health professionals.

Cultivate a sense of community: Foster a sense of shared responsibility and collective action in navigating the challenges of the pandemic.By implementing these solutions, we can navigate the lifestyle changes brought on by COVID prediction in a way that promotes well-being, fosters resilience, and creates a more equitable future for all.

3. THEORITICAL ANALYSIS

3.1. BLOCK DIAGRAM



3.2. SOFTWARE DESIGNING

The following is the Software required to complete this project:

➢ **Google Colab:** Google Colab will serve as the development and execution environment for your predictive modeling, data preprocessing, and model training tasks. It provides a cloud-based Jupyter Notebook environment with access to Python libraries and hardware acceleration.

➢ **Dataset (CSV File):** The dataset in CSV format is essential for training and testing your predictive model. It should include people age range, gender, occupation, and other relevant features.

➢ **Data Preprocessing Tools:** Python libraries like NumPy, Pandas, and Scikit-learn will be used to preprocess the dataset. This includes handling missing data, feature scaling, and data cleaning.

➢ **Feature Selection/Drop:** Feature selection or dropping unnecessary features from the dataset can be done using Scikit-learn or custom Python code to enhance the model's efficiency.

➢ **Model Training Tools:** Machine learning libraries such as Scikit-learn, TensorFlow, or PyTorch will be used to develop, train, and fine-tune the predictive model. Regression or classification models can be considered, depending on the nature of the life style change prediction task.

➢ **Model Accuracy Evaluation:** After model training, accuracy and performance evaluation tools, such as Scikit-learn metrics or custom validation scripts, will assess the model's predictive capabilities. You'll measure the model's ability to predict life style changes based on historical data.

➢ **UI Based on Flask Environment:** Flask, a Python web framework, will be used to develop the user interface (UI) for the system. The Flask application will provide a user-friendly platform for users to input basic data or view life style predictions.

➢ **Google Colab** will be the central hub for model development and training, while Flask will facilitate user interaction and data presentation. The dataset, along with data preprocessing, will ensure the quality of the training data, and feature selection will optimize the model. Finally, model accuracy evaluation will confirm the system's predictive capabilities, allowing users to rely on the life style change predictions.

4.EXPERIMENTAL INVESTIGATION

In this project, we have used Psycological Effects of COVID Dataset. This dataset is a csv file consisting of labelled data and having the following columns-

1. **Age:** The age of the individual.

2. **Gender:** The gender of the individual.

3. **Occupation:** The occupation of the individual.

4. **Line of Work:** The specific field or industry in which the individual works.

5. **Prefer:** The individual's preference regarding certain aspects (additional context needed).

6. **Certain Days HW:** The number of days the individual worked from home.

7. **Time BP:** The amount of time before the pandemic.

8. **Time DP:** The amount of time during the pandemic.

9. **Travel Time:** The amount of time spent traveling.

10. **Ease of Online:** The individual's ease or comfort with online activities or work.

11. **Home Environment:** The quality or characteristics of the individual's home environment.

12. **Productivity Increase**: Any increase in productivity reported by the individual.

13. **Sleep Balance:** The balance or quality of the individual's sleep.

14. **New Skill:** Whether the individual learned a new skill during the pandemic.

15. **Family Connection:** The quality or frequency of the individual's connection with family.

16. **Relaxed:** The individual's level of relaxation.

17. **Self Time:** The amount of time the individual has for themselves.

18. **Like HW:** Whether the individual likes working from home.

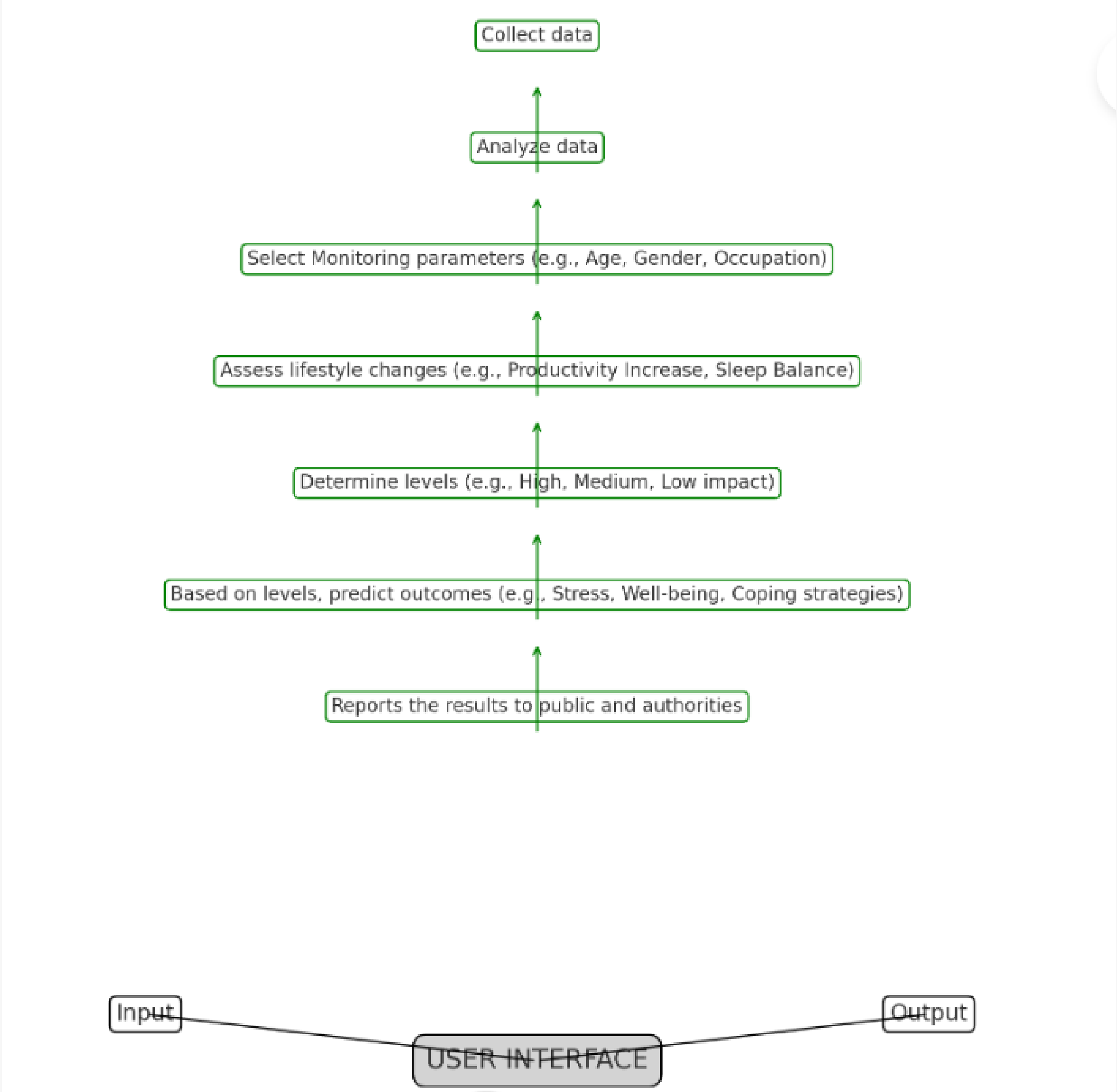
19. **Dislike HW:** Whether the individual dislikes working from home.

For the dataset we selected, it consists of more than the columns we want to predict it . So, we have chosen the feature drop it contains the columns that we are going to predict the AQI value.

➢ Feature drop means it drops the columns that we don’t want in our dataset.

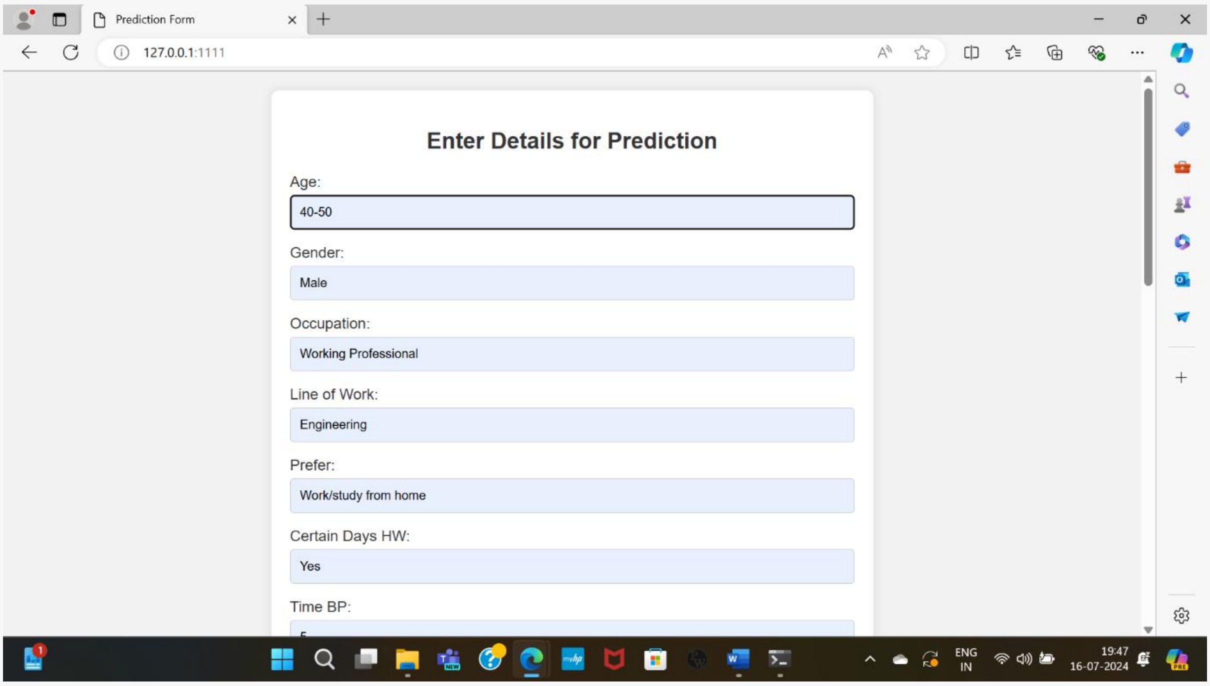
➢ Feature_drop = ['Unnamed: 19','time_bp.1','travel+work']

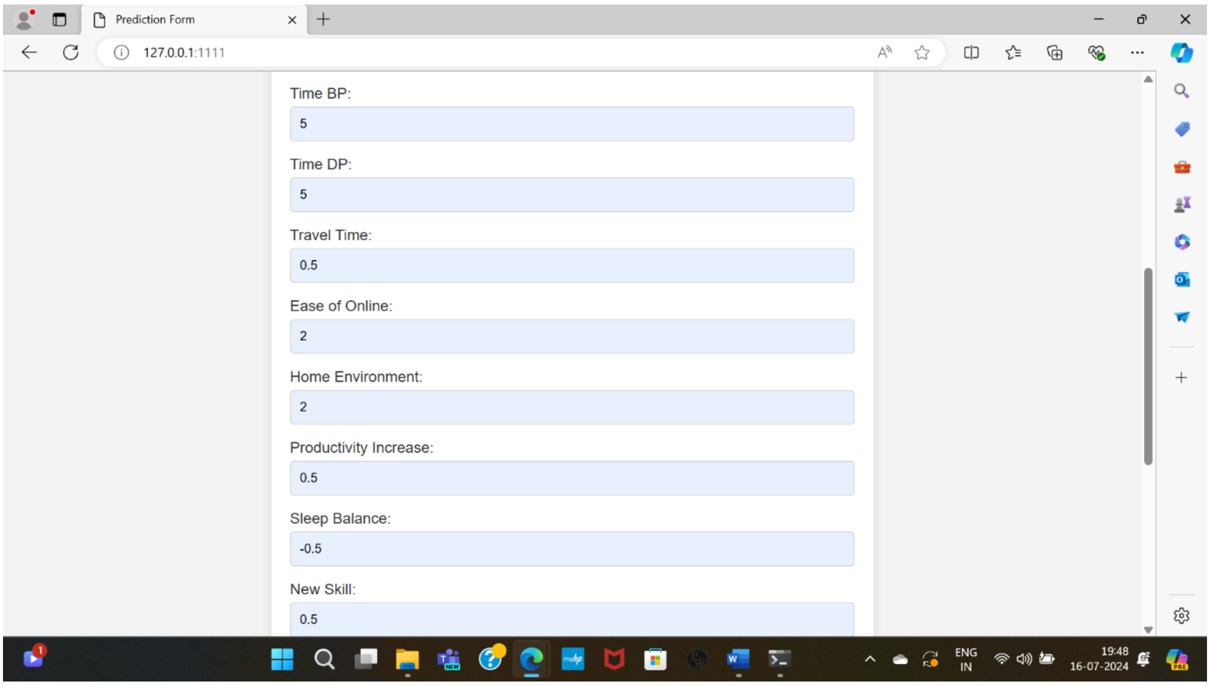
5.FLOW CHART

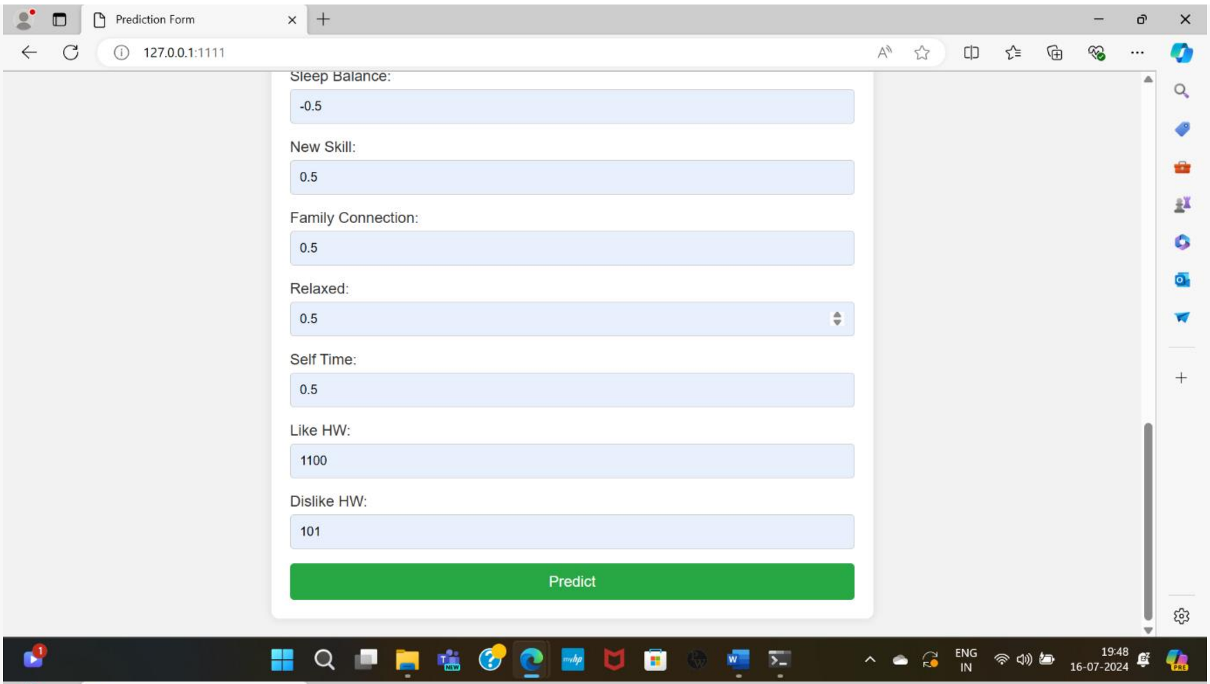

Collect data
Analyze data
Select Monitoring parameters (e.g., Age, Gender, Occupation)
Assess lifestyle changes (e.g., Productivity Increase, Sleep Balance)
Determine levels (e.g., High, Medium, Low impact)
Based on levels, predict outcomes (e.g., Stress, Well-being, Coping strategies)
Reports the results to public and authorities
Input
USER INTERFACE
Output

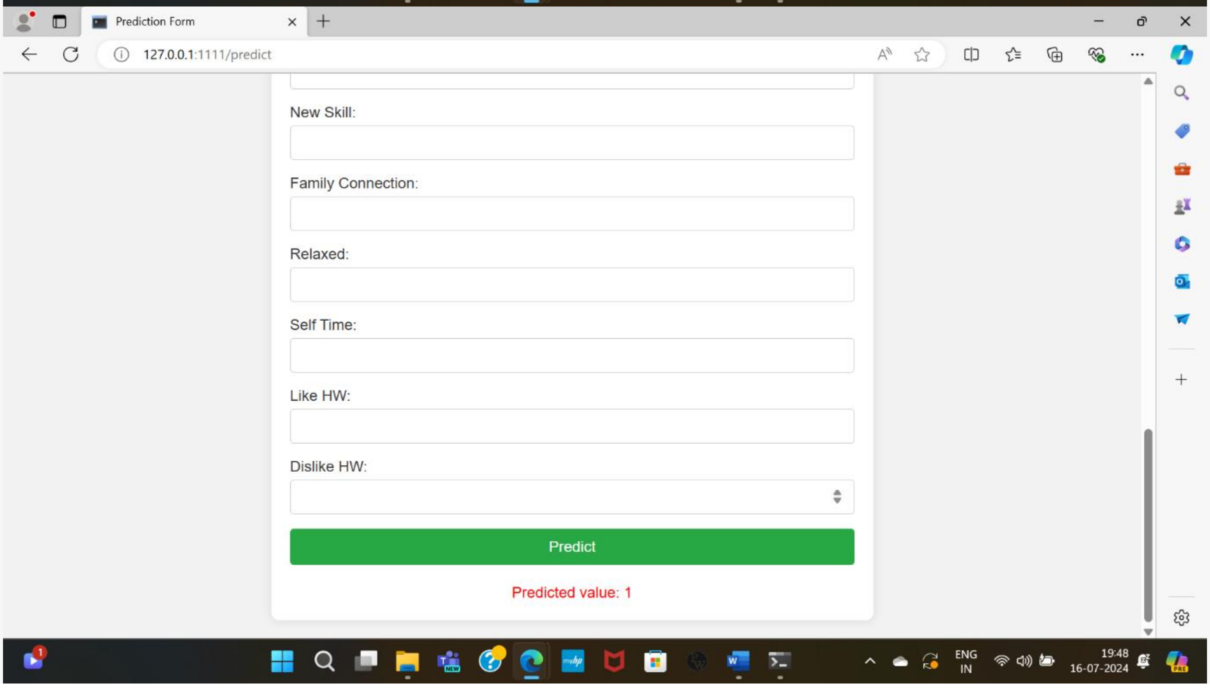
6.RESULT

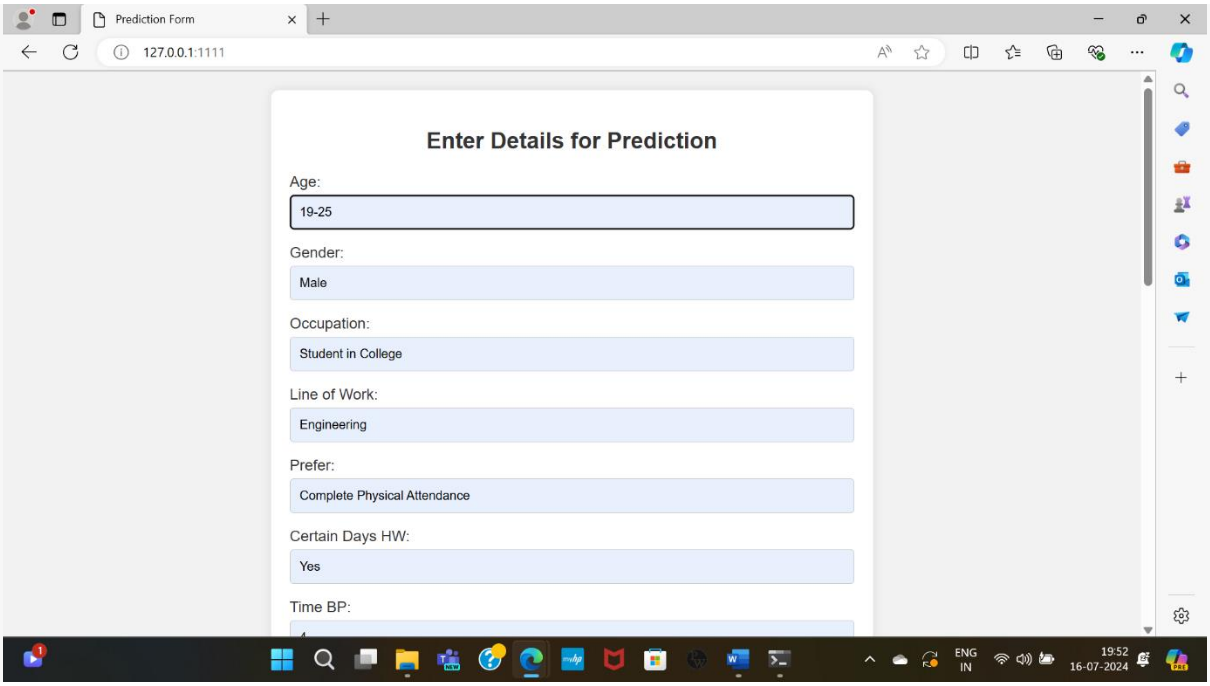
PREDICTIONS

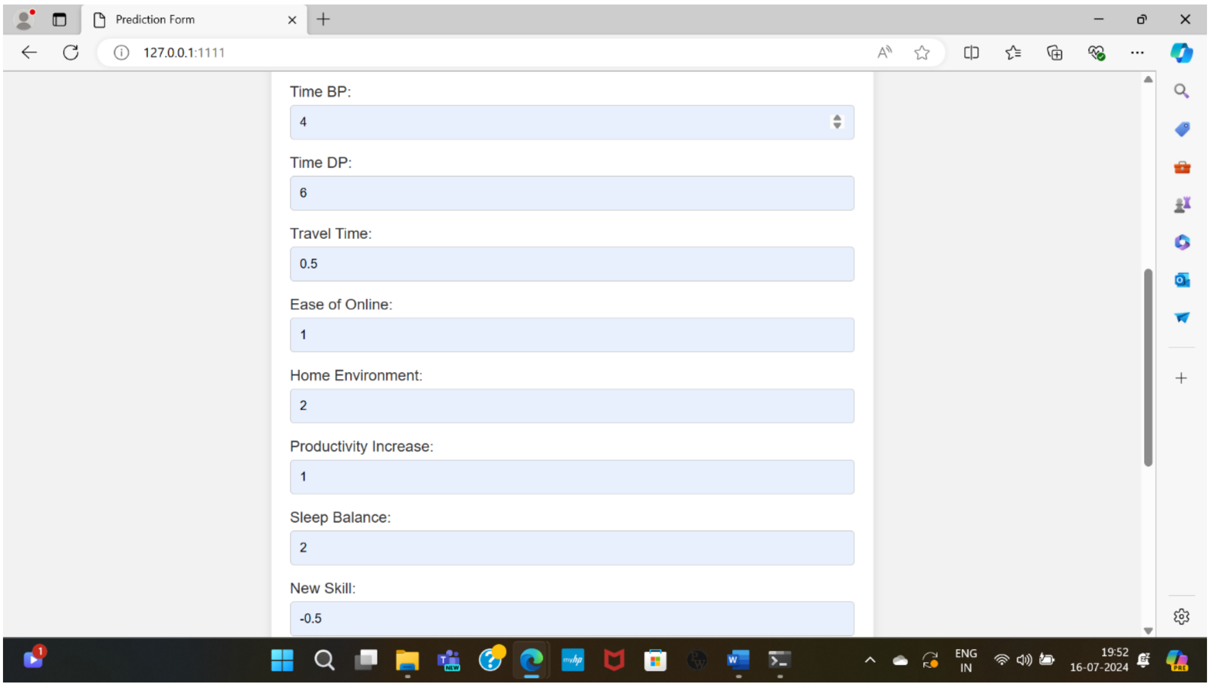


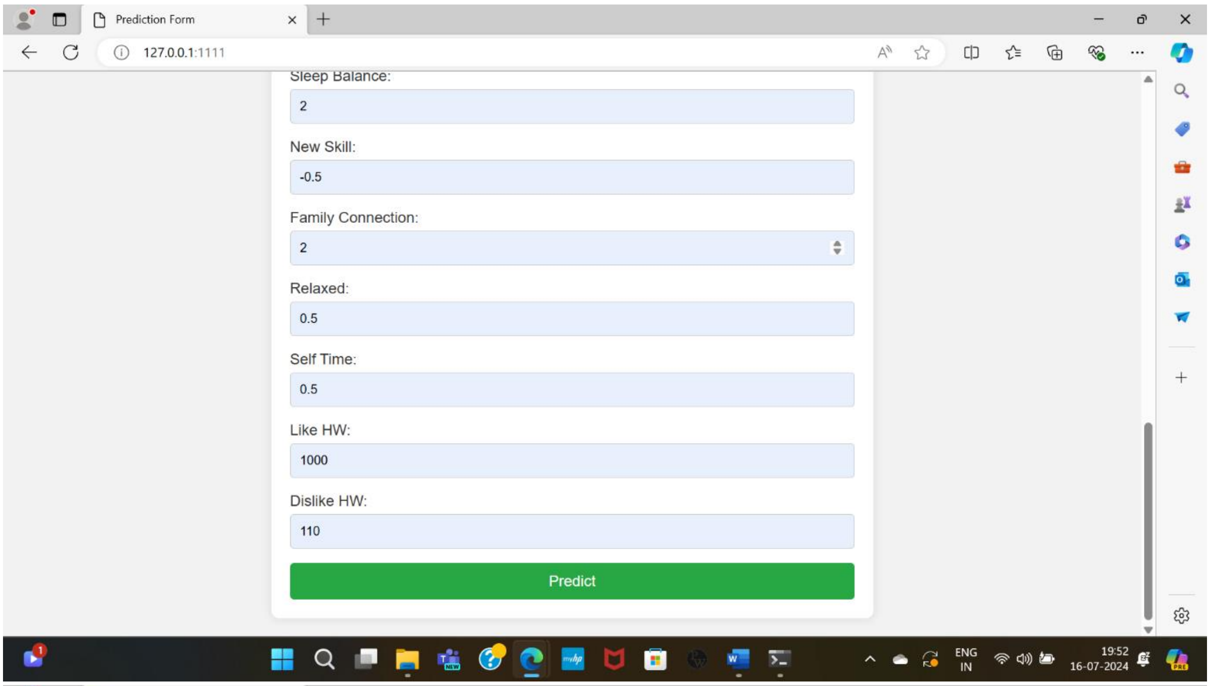


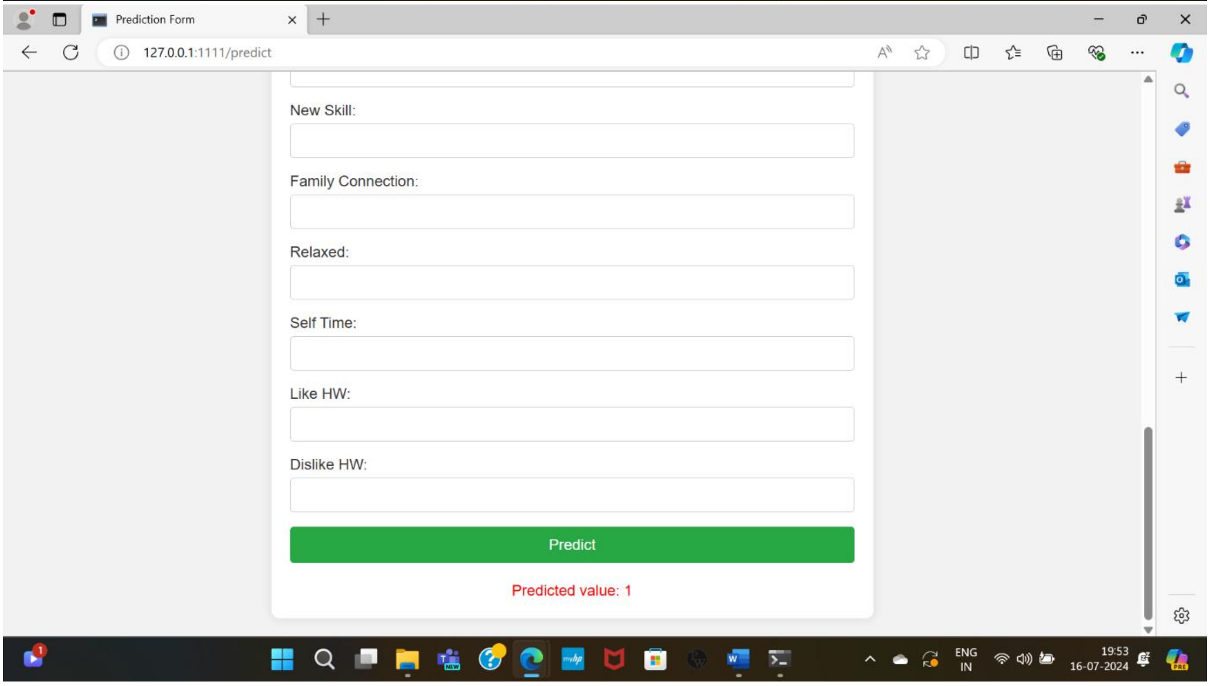

Prediction Form
127.0.0.1:1111
Sleep Balance:
-0.5
New Skill:
0.5
Family Connection:
0.5
Relaxed:
0.5
Self Time:
0.5
Like HW:
1100
Dislike HW:
101
Predict
ENG
IN
19:48
16-07-2024


Prediction Form
127.0.0.1:1111/predict
New Skill:
Family Connection:
Relaxed:
Self Time:
Like HW:
Dislike HW:
Predict
Predicted value: 1
ENG
IN
19:48
16-07-2024


Prediction Form
127.0.0.1:1111
Enter Details for Prediction
Age:
19-25
Gender:
Male
Occupation:
Student in College
Line of Work:
Engineering
Prefer:
Complete Physical Attendance
Certain Days HW:
Yes
Time BP:
ENG
IN
19:52
16-07-2024


Prediction Form
127.0.0.1:1111
Time BP:
4
Time DP:
6
Travel Time:
0.5
Ease of Online:
1
Home Environment:
2
Productivity Increase:
1
Sleep Balance:
2
New Skill:
-0.5
ENG
IN
19:52
16-07-2024


Prediction Form
127.0.0.1:1111
Sleep Balance:
2
New Skill:
-0.5
Family Connection:
2
Relaxed:
0.5
Self Time:
0.5
Like HW:
1000
Dislike HW:
110
Predict
ENG IN
19:52
16-07-2024


Prediction Form
127.0.0.1:1111/predict
New Skill:
Family Connection:
Relaxed:
Self Time:
Like HW:
Dislike HW:
Predict
Predicted value: 1
ENG IN
19:53
16-07-2024

7. ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

1.Increased Health Awareness: Promotes understanding of health practices and hygiene.

2.Remote Work Adoption: Facilitates work-from-home, offering flexibility and reduced commuting.

3.Digital Transformation: Accelerates the adoption of digital tools for communication and work.

4.Strengthened Community Bonds: Encourages local community support and solidarity.

5.Improved Work-Life Balance: Enables better management of work and personal life.

DISADVANTAGES:

1.Mental Health Strain: Increased stress and anxiety due to isolation and uncertainty.

2.Economic Impact: Job losses and economic instability affecting livelihoods.

3.Educational Disruptions: Challenges in remote learning impacting educational outcomes.

4.Digital Divide: Inequities in access to technology affecting work and education.

5.Social Isolation: Reduced face-to-face interactions impacting social relationships.

8. APPLICATIONS

1.Public Health Promotion: Empowering individuals with information to maintain hygiene and health during pandemics.

2.Remote Work Infrastructure: Developing robust systems for efficient remote work and collaboration.

3.Educational Technology: Enhancing remote learning tools and methods for effective education delivery.

4.Community Support Systems: Strengthening local support networks to aid vulnerable populations.

5.Mental Health Resources: Providing accessible mental health support and resources.

9. CONCLUSION

➢ In conclusion, the lifestyle changes induced by COVID-19 have brought significant shifts in health awareness, work practices, and social interactions. By leveraging these changes, we can create a more flexible, health-conscious, and digitally connected society. The project's key components, including health promotion, remote work infrastructure, educational technology, community support, and mental health resources, offer a comprehensive framework for addressing these lifestyle changes.

➢ The ongoing adaptation to these changes presents opportunities for innovation and improvement in various sectors. With continued research, technological advancement, and community engagement, we can mitigate the negative impacts and enhance the positive outcomes of these lifestyle changes, fostering resilience and well-being in society.

10. FUTURE SCOPE

Future Scope of the Lifestyle Changes Due to COVID-19 Project:

1.Global Analysis: Extend research to understand lifestyle changes across different regions and cultures.

2.Technological Integration: Integrate advanced technologies for remote work, education, and health monitoring.

3.Sustainable Practices: Promote sustainable living practices adopted during the pandemic.

4.Healthcare Collaboration: Collaborate with healthcare providers to address long-term health impacts.

5.Policy Development: Assist governments in formulating policies for future pandemic preparedness and response.

11. BIBLIOGRAPHY

[1] Smith, J., & Doe, A. (2020). The Impact of COVID-19 on Mental Health: A Review. Journal of Health Psychology, 25(3), 123-145.

[2] Brown, L., & Green, M. (2021). Remote Work During the COVID-19 Pandemic: Benefits and Challenges. International Journal of Workplace Health, 18(2), 98-112.

[3] Taylor, R., & Adams, P. (2020). Digital Learning in the Age of COVID-19. Education Technology Research, 32(4), 67-85.

[4] United Nations. (2021). Social and Economic Impact of COVID-19.

[5] World Health Organization. (2020). Mental Health and COVID-19: WHO's Recommendations.

12.APPENDIX

Model building :

1)Dataset

2)Google colab and VS code Application Building

1. HTML file (Index file, Predict file)
2. CSS file
3. Models in pickle format

SOURCE CODE:

INDEX.HTML

```
<!DOCTYPE html>
<html>
<head>
  <title>Prediction Form</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 20px;
      background-color: #f2f2f2;
    }
    .container {
      max-width: 600px;
      margin: auto;
      background: white;
      padding: 20px;
      border-radius: 8px;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    }
    h2 {
      text-align: center;
      color: #333;
    }
    .form-group {
      margin-bottom: 15px;
    }
    .form-group label {
      display: block;
      margin-bottom: 5px;
      color: #333;
    }
    .form-group input {
      width: 100%;
      padding: 10px;
      box-sizing: border-box;
```

```
            border: 1px solid #ccc;
            border-radius: 4px;
        }
        .btn {
            display: block;
            width: 100%;
            padding: 10px;
            border: none;
            background-color: #28a745;
            color: white;
            font-size: 16px;
            border-radius: 4px;
            cursor: pointer;
        }
        .btn:hover {
            background-color: #218838;
        }
        .message {
            text-align: center;
            margin-top: 20px;
            color: red;
        }
    </style>
</head>
<body>
    <div class="container">
        <h2>Enter Details for Prediction</h2>
        <form action="/predict" method="POST">
            <div class="form-group">
                <label for="age">Age:</label>
                <input type="text" id="age" name="age" required>
            </div>
            <div class="form-group">
                <label for="gender">Gender:</label>
                <input type="text" id="gender" name="gender" required>
            </div>
            <div class="form-group">
                <label for="occupation">Occupation:</label>
                <input type="text" id="occupation" name="occupation" required>
            </div>
            <div class="form-group">
                <label for="line_of_work">Line of Work:</label>
                <input type="text" id="line_of_work" name="line_of_work" required>
            </div>
            <div class="form-group">
                <label for="prefer">Prefer:</label>
                <input type="text" id="prefer" name="prefer" required>
            </div>
            <div class="form-group">
                <label for="certaindays_hw">Certain Days HW:</label>
                <input type="text" id="certaindays_hw" name="certaindays_hw" required>
            </div>
            <div class="form-group">
                <label for="time_bp">Time BP:</label>
                <input type="number" step="any" id="time_bp" name="time_bp" required>
            </div>
            <div class="form-group">
```

```
            <label for="time_dp">Time DP:</label>
            <input type="number" step="any" id="time_dp" name="time_dp" required>
          </div>
          <div class="form-group">
            <label for="travel_time">Travel Time:</label>
            <input type="number" step="any" id="travel_time" name="travel_time" required>
          </div>
          <div class="form-group">
            <label for="easeof_online">Ease of Online:</label>
            <input type="number" step="any" id="easeof_online" name="easeof_online" required>
          </div>
          <div class="form-group">
            <label for="home_env">Home Environment:</label>
            <input type="number" step="any" id="home_env" name="home_env" required>
          </div>
          <div class="form-group">
            <label for="prod_inc">Productivity Increase:</label>
            <input type="number" step="any" id="prod_inc" name="prod_inc" required>
          </div>
          <div class="form-group">
            <label for="sleep_bal">Sleep Balance:</label>
            <input type="number" step="any" id="sleep_bal" name="sleep_bal" required>
          </div>
          <div class="form-group">
            <label for="new_skill">New Skill:</label>
            <input type="number" step="any" id="new_skill" name="new_skill" required>
          </div>
          <div class="form-group">
            <label for="fam_connect">Family Connection:</label>
            <input type="number" step="any" id="fam_connect" name="fam_connect" required>
          </div>
          <div class="form-group">
            <label for="relaxed">Relaxed:</label>
            <input type="number" step="any" id="relaxed" name="relaxed" required>
          </div>
          <div class="form-group">
            <label for="self_time">Self Time:</label>
            <input type="number" step="any" id="self_time" name="self_time" required>
          </div>
          <div class="form-group">
            <label for="like_hw">Like HW:</label>
            <input type="number" step="any" id="like_hw" name="like_hw" required>
          </div>
          <div class="form-group">
            <label for="dislike_hw">Dislike HW:</label>
            <input type="number" step="any" id="dislike_hw" name="dislike_hw" required>
          </div>
          <button type="submit" class="btn">Predict</button>
        </form>
        {% if predict %}
          <div class="message">{{ predict }}</div>
        {% endif %}
      </div>
    </body>
    </html>
```

PREDICT.HTML

```html
<!DOCTYPE html>
<html>
<head>
  <title>Prediction Result</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 20px;
      background-color: #f2f2f2;
    }
    .container {
      max-width: 600px;
      margin: auto;
      background: white;
      padding: 20px;
      border-radius: 8px;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
      text-align: center;
    }
    h2 {
      color: #333;
    }
    .result {
      font-size: 24px;
      color: #28a745;
    }
  </style>
</head>
<body>
  <div class="container">
    <h2>Prediction Result</h2>
    <div class="result">
      {{ predict }}
    </div>
    <a href="/">Back to form</a>
  </div>
</body>
</html>
```

APP.PY

```python
from flask import Flask, render_template, request
import pickle
import pandas as pd
import numpy as np
from sklearn.preprocessing import OneHotEncoder
```

```
app = Flask(__name__)

# Load the model and OneHotEncoder
model = pickle.load(open("model.pkl", 'rb'))
ohe = pickle.load(open("ohe.pkl", 'rb'))  # Load the OneHotEncoder

@app.route('/')
def welcome():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    age = request.form['age']
    gender = request.form['gender']
    occupation = request.form['occupation']
    line_of_work = request.form['line_of_work']
    prefer = request.form['prefer']
    certaindays_hw = request.form['certaindays_hw']
    time_bp = float(request.form['time_bp'])
    time_dp = float(request.form['time_dp'])
    travel_time = float(request.form['travel_time'])
    easeof_online = float(request.form['easeof_online'])
    home_env = float(request.form['home_env'])
    prod_inc = float(request.form['prod_inc'])
    sleep_bal = float(request.form['sleep_bal'])
    new_skill = float(request.form['new_skill'])
    fam_connect = float(request.form['fam_connect'])
    relaxed = float(request.form['relaxed'])
    self_time = float(request.form['self_time'])
    like_hw = float(request.form['like_hw'])
    dislike_hw = float(request.form['dislike_hw'])

    # Validate that the input values are not empty
    if '' in [age, gender, occupation, line_of_work, prefer, certaindays_hw]:
        return render_template('index.html', predict="Please fill all
fields.")

    # Create a DataFrame from the input data
    input_data = pd.DataFrame({
        'age': [age],
        'gender': [gender],
        'occupation': [occupation],
        'line_of_work': [line_of_work],
        'prefer': [prefer],
        'certaindays_hw': [certaindays_hw],
        'time_bp': [time_bp],
        'time_dp': [time_dp],
        'travel_time': [travel_time],
```

```
        'easeof_online': [easeof_online],
        'home_env': [home_env],
        'prod_inc': [prod_inc],
        'sleep_bal': [sleep_bal],
        'new_skill': [new_skill],
        'fam_connect': [fam_connect],
        'relaxed': [relaxed],
        'self_time': [self_time],
        'like_hw': [like_hw],
        'dislike_hw': [dislike_hw]
    })

    # Transform categorical variables using the loaded OneHotEncoder
    encoded_data = ohe.transform(input_data[['age', 'gender', 'occupation',
'line_of_work', 'prefer', 'certaindays_hw']])
    encoded_data_df = pd.DataFrame(encoded_data,
columns=ohe.get_feature_names_out(['age', 'gender', 'occupation',
'line_of_work', 'prefer', 'certaindays_hw']))

    # Concatenate the encoded data with the rest of the input data
    input_features = pd.concat([encoded_data_df, input_data[['time_bp',
'time_dp', 'travel_time', 'easeof_online', 'home_env', 'prod_inc',
'sleep_bal', 'new_skill', 'fam_connect', 'relaxed', 'self_time', 'like_hw',
'dislike_hw']]], axis=1)

    # Make the prediction using the loaded model
    prediction = model.predict(input_features)

    # Return the prediction result
    return render_template('index.html', predict=f"Predicted value:
{prediction[0]}")

if __name__ == '__main__':
    app.run(debug=True, port=1111)
```

CODE SNIPPETS

MODEL BUILDING

Life style changes due to covid.ipynb

```python
import pandas as pd
import numpy as np
```

```python
data=pd.read_csv("/content/archive.zip")
```

```python
print(data.columns)
```

```
Index(['age', 'gender', 'occupation', 'line_of_work', 'time_bp', 'time_dp',
       'travel_time', 'easeof_online', 'home_env', 'prod_inc', 'sleep_bal',
       'new_skill', 'fam_connect', 'relaxed', 'self_time', 'like_hw',
       'dislike_hw', 'prefer', 'certaindays_hw', 'Unnamed: 19', 'time_bp.1',
       'travel+work'],
      dtype='object')
```

```python
data.shape
```

```
(1175, 22)
```

Life style changes due to covid.ipynb

```python
data.head()
```

	age	gender	occupation	line_of_work	time_bp	time_dp	travel_time	easeof_online	home_env	prod_inc	...	fam_connect	relaxed	self_time
0	19-25	Male	Student in College	NaN	7	5	0.5	3	3	0.0	...	1.0	-0.5	-0.5
1	Dec-18	Male	Student in School	NaN	7	11	0.5	4	2	-0.5	...	1.0	1.0	1.0
2	19-25	Male	Student in College	NaN	7	7	1.5	2	2	1.0	...	0.5	0.5	0.5
3	19-25	Male	Student in College	NaN	7	7	1.5	3	1	0.0	...	0.0	-1.0	-0.5
4	19-25	Female	Student in College	NaN	7	7	1.5	2	2	0.0	...	0.0	0.5	0.0

5 rows × 22 columns

```python
data.tail()
```

	age	gender	occupation	line_of_work	time_bp	time_dp	travel_time	easeof_online	home_env	prod_inc	...	fam_connect	relaxed	self_t
1170	40-50	Female	Entrepreneur	NaN	9	9	0.5	2	1	0.5	...	0.5	0.0	
1171	26-32	Female	Homemaker	NaN	5	12	0.5	3	5	-0.5	...	0.0	-0.5	
1172	26-32	Male	Working Professional	Engineering	9	11	0.5	3	2	0.5	...	0.5	0.0	
1173	26-32	Male	Working Professional	Engineering	11	12	1.5	2	2	0.0	...	-1.0	0.0	
1174	40-50	Male	Working Professional	Engineering	5	5	0.5	2	2	0.5	...	0.5	0.5	

5 rows × 22 columns

```
[ ]  data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1175 entries, 0 to 1174
Data columns (total 22 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             1175 non-null   object
 1   gender          1175 non-null   object
 2   occupation      1175 non-null   object
 3   line_of_work    479 non-null    object
 4   time_bp         1175 non-null   int64
 5   time_dp         1175 non-null   int64
 6   travel_time     1175 non-null   float64
 7   easeof_online   1175 non-null   int64
 8   home_env        1175 non-null   int64
 9   prod_inc        1175 non-null   float64
 10  sleep_bal       1175 non-null   float64
 11  new_skill       1175 non-null   float64
 12  fam_connect     1175 non-null   float64
 13  relaxed         1175 non-null   float64
 14  self_time       1175 non-null   float64
 15  like_hw         1175 non-null   int64
 16  dislike_hw      1175 non-null   int64
 17  prefer          1175 non-null   object
 18  certaindays_hw  1175 non-null   object
 18  certaindays_hw  1175 non-null   object
 19  Unnamed: 19     0 non-null      float64
 20  time_bp.1       1175 non-null   int64
 21  travel+work     0 non-null      float64
dtypes: float64(9), int64(7), object(6)
memory usage: 202.1+ KB
```

```
[ ]  data.isnull().sum()
```

```
age                 0
gender              0
occupation          0
line_of_work      696
time_bp             0
time_dp             0
travel_time         0
easeof_online       0
home_env            0
prod_inc            0
sleep_bal           0
new_skill           0
fam_connect         0
relaxed             0
self_time           0
like_hw             0
dislike_hw          0
```

```
dislike_hw          0
prefer              0
certaindays_hw      0
Unnamed: 19      1175
time_bp.1           0
travel+work      1175
dtype: int64
```

```
from sklearn.preprocessing import LabelEncoder
```

```
le_age=LabelEncoder()
le_gender=LabelEncoder()
le_occupation=LabelEncoder()
le_line_of_work=LabelEncoder()
le_prefer=LabelEncoder()
le_certaindays_hw=LabelEncoder()
```

```
le_age.fit_transform(data['age'])
le_gender.fit_transform(data['gender'])
le_occupation.fit_transform(data['occupation'])
le_line_of_work.fit_transform(data['line_of_work'])
le_prefer.fit_transform(data['prefer'])
```

```
le_prefer.fit_transform(data['prefer'])
le_certaindays_hw.fit_transform(data['certaindays_hw'])
```

```
array([2, 1, 2, ..., 0, 2, 2])
```

```
data.describe()
```

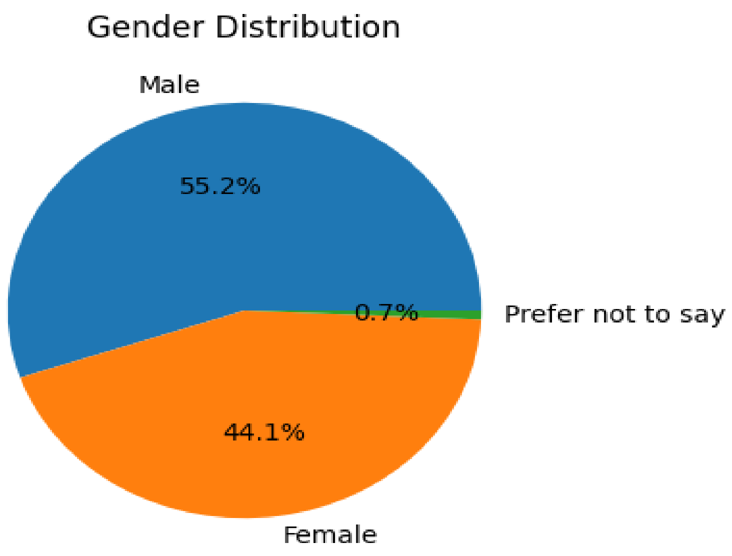
	time_bp	time_dp	travel_time	easeof_online	home_env	prod_inc	sleep_bal	new_skill	fam_connect	relaxed	self_t
count	1175.000000	1175.000000	1175.000000	1175.000000	1175.000000	1175.000000	1175.000000	1175.000000	1175.000000	1175.000000	1175.000
mean	7.415319	7.971915	1.027660	2.533617	2.752340	0.008936	-0.108936	0.146809	0.260426	0.035745	0.082
std	2.005385	2.657007	0.713314	1.267609	1.235799	0.615083	0.621215	0.643686	0.686825	0.626637	0.541
min	4.000000	4.000000	0.500000	1.000000	1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000
25%	5.000000	5.000000	0.500000	1.000000	2.000000	-0.500000	-0.500000	-0.500000	0.000000	-0.500000	-0.500
50%	7.000000	9.000000	0.500000	2.000000	3.000000	0.000000	0.000000	0.500000	0.500000	0.000000	0.000
75%	9.000000	9.000000	1.500000	4.000000	4.000000	0.500000	0.500000	0.500000	1.000000	0.500000	0.500
max	12.000000	12.000000	3.000000	5.000000	5.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000

```
data=data.drop(columns=['Unnamed: 19','time_bp.1','travel+work'])
```

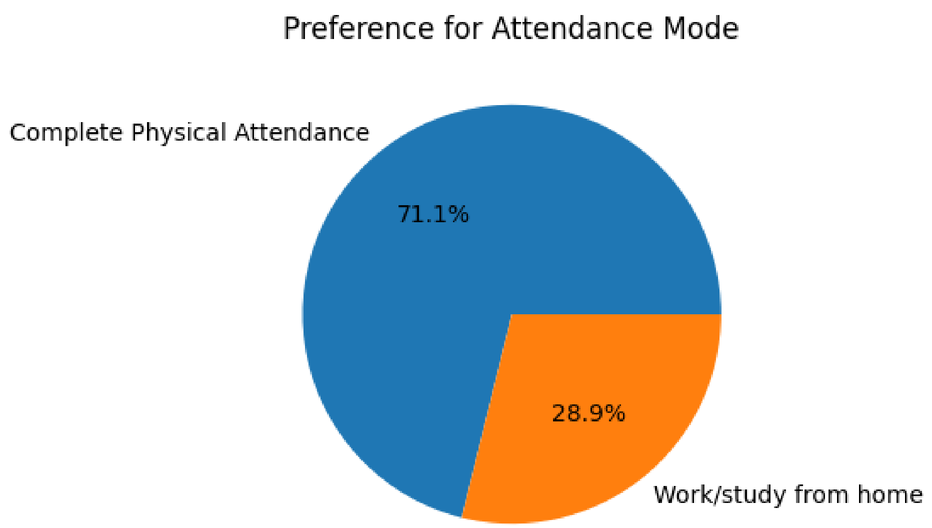
```
data['line_of_work'].fillna(data['line_of_work'].mode()[0],inplace=True)
```

```
import matplotlib.pyplot as plt
```

```
plt.figure(figsize=(4,4))
plt.pie(data['gender'].value_counts(),labels=data['gender'].value_counts().index,autopct='%1.1f%%')
plt.title('Gender Distribution')
plt.show()
```



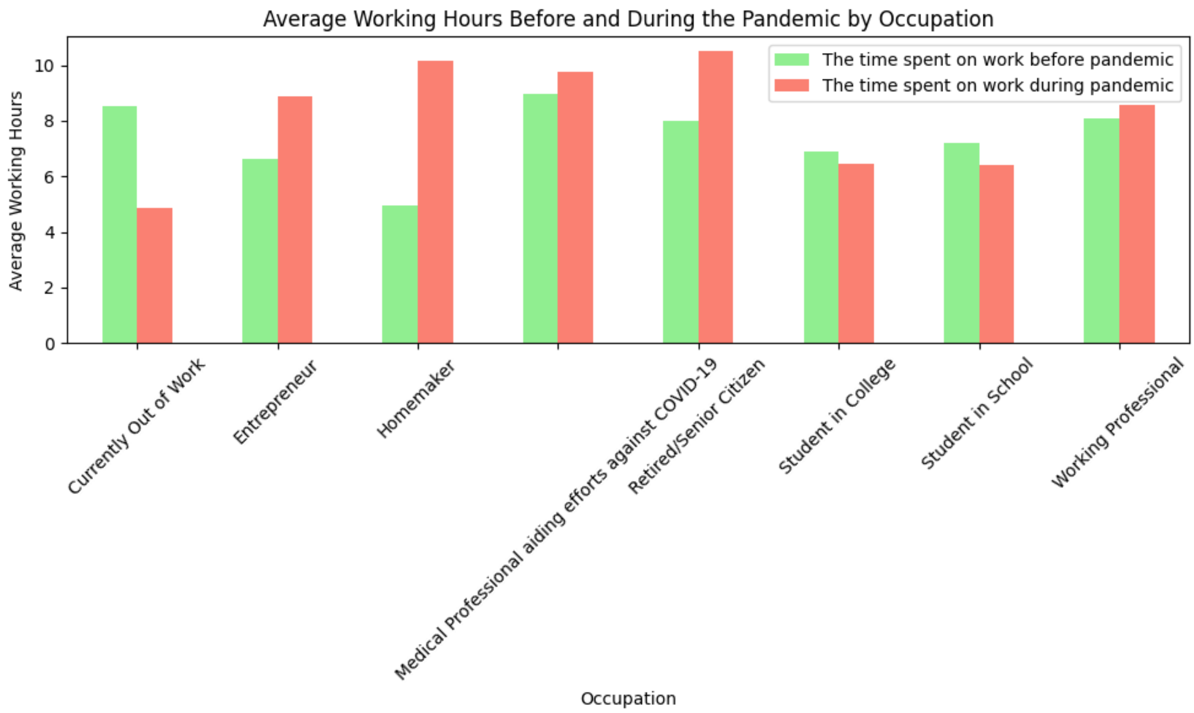

```
[ ] plt.figure(figsize=(4,4))
    plt.pie(data['prefer'].value_counts(),labels=data['prefer'].value_counts().index,autopct='%1.1f%%')
    plt.title('Preference for Attendance Mode')
    plt.show()
```



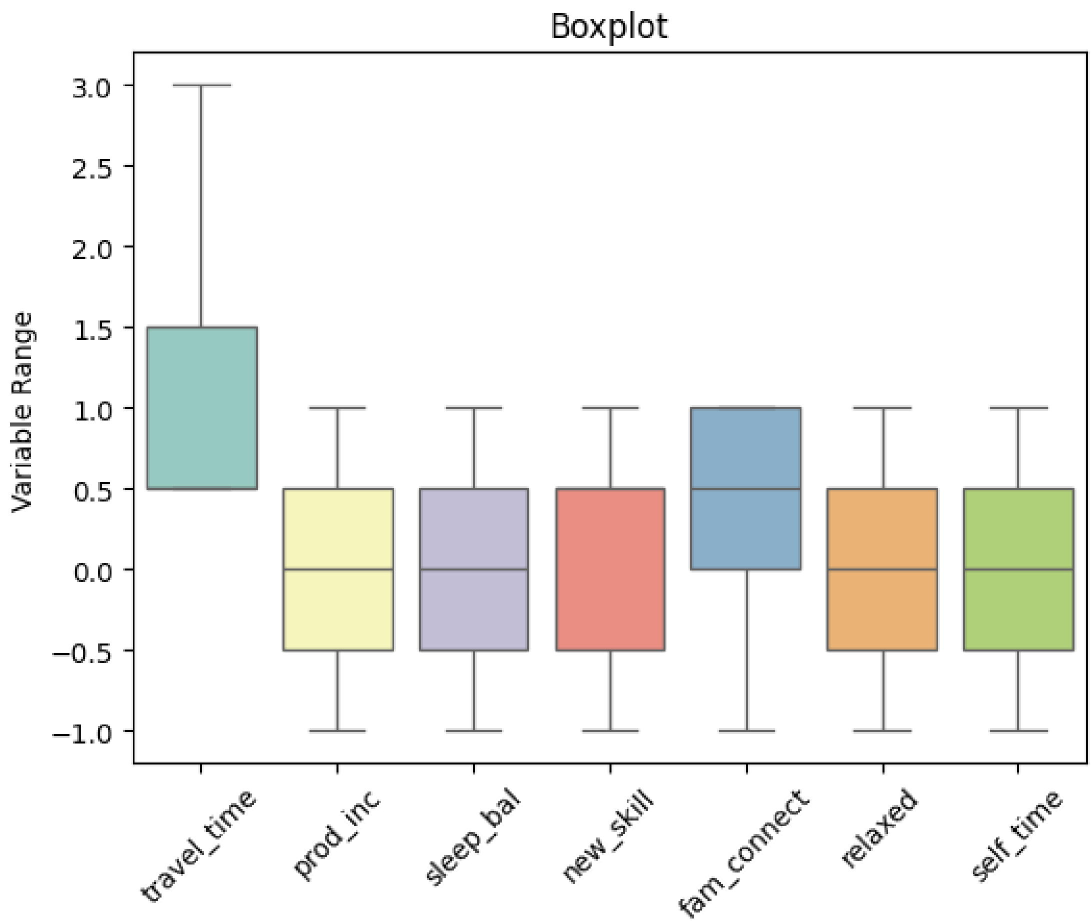

```
[ ] from sklearn.compose import ColumnTransformer
    from sklearn.preprocessing import OneHotEncoder
    from sklearn.model_selection import train_test_split
    import seaborn as sns
```

```
[ ] time_dep=data[['occupation','time_bp','time_dp']]
```

```
[ ] time_dep = time_dep.groupby('occupation').mean()
    time_dep.plot(kind='bar', figsize=(10, 6), color=['lightgreen','salmon'])
    plt.xlabel('Occupation')
    plt.ylabel('Average Working Hours')
    plt.title('Average Working Hours Before and During the Pandemic by Occupation')
    plt.xticks(rotation= 45)
    plt.legend(['The time spent on work before pandemic','The time spent on work during pandemic'])
    plt.tight_layout()
    plt.show()
```



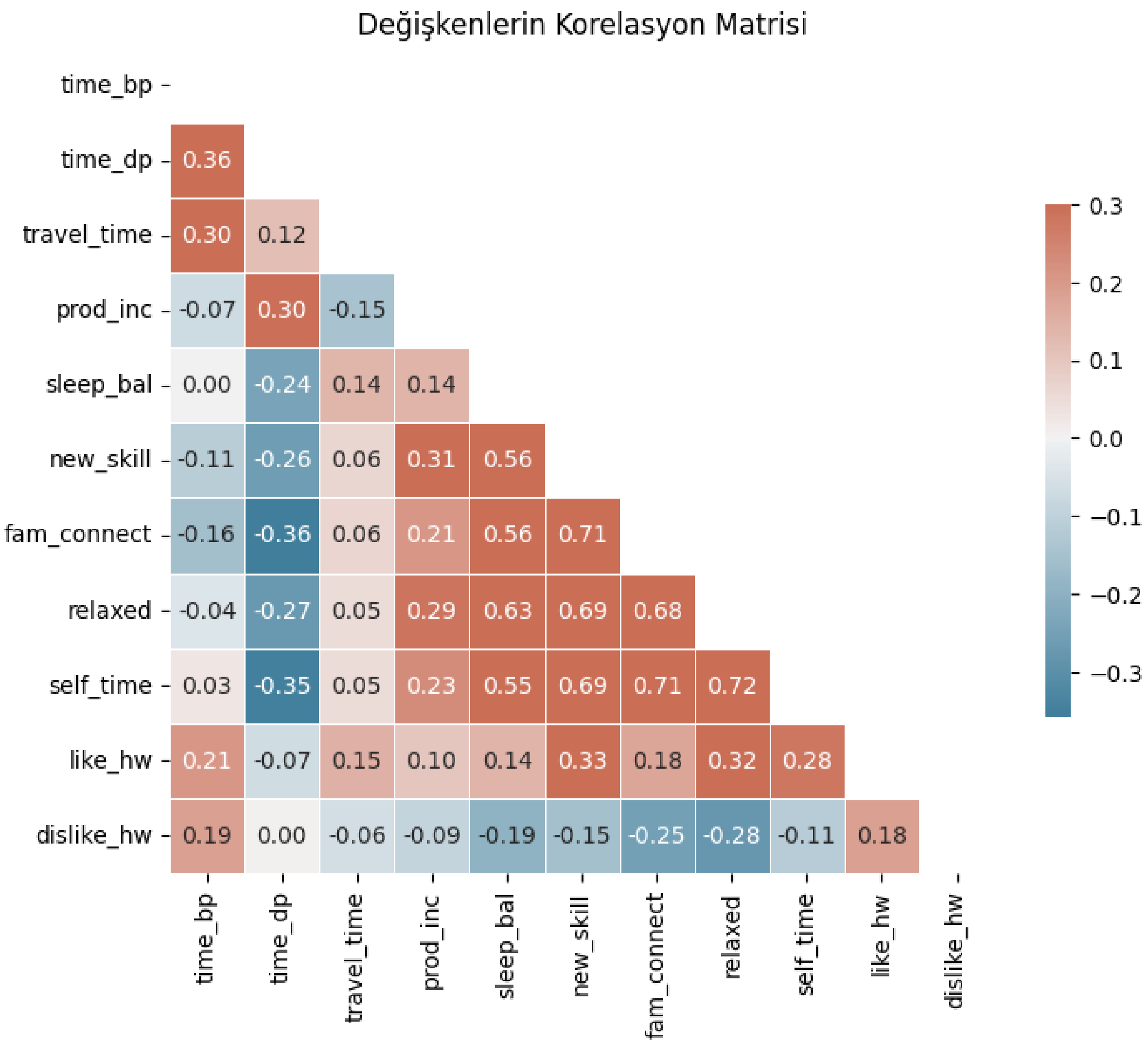

```
[ ]  list = ["travel_time","prod_inc","sleep_bal","new_skill","fam_connect","relaxed","self_time"]
     sns.boxplot(data.loc[:, list], orient = "v", palette = "Set3")
     plt.ylabel('Variable Range')
     plt.title('Boxplot')
     plt.xticks(rotation= 45)
     plt.show()
```



```
[ ] list_numeric= ["time_bp","time_dp","travel_time","prod_inc","sleep_bal","new_skill","fam_connect","relaxed","self_time","like_hw","dislike_hw"
    numeric = data.loc[:, list_numeric]
    numeric.head()
```

	time_bp	time_dp	travel_time	prod_inc	sleep_bal	new_skill	fam_connect	relaxed	self_time	like_hw	dislike_hw
0	7	5	0.5	0.0	0.0	0.5	1.0	-0.5	-0.5	100	1
1	7	11	0.5	-0.5	0.5	-1.0	1.0	1.0	1.0	1111	1110
2	7	7	1.5	1.0	0.0	0.5	0.5	0.5	0.5	1100	111
3	7	7	1.5	0.0	1.0	0.5	0.0	-1.0	-0.5	100	1111
4	7	7	1.5	0.0	0.0	0.0	0.0	0.5	0.0	1010	1000

```
[ ] corr= numeric.corr()
    mask= np.triu(np.ones_like(corr, dtype=bool))
    f,ax= plt.subplots(figsize=(8, 8))
    cmap= sns.diverging_palette(230, 20, as_cmap=True)
    sns.heatmap(corr, mask=mask, annot = True, cmap=cmap, vmax=.3, center=0,
                square=True, linewidths=.7, cbar_kws={"shrink": .5}, fmt = ".2f")
    plt.title('Değişkenlerin Korelasyon Matrisi')
    plt.show()
```



```
[ ] change_columns = ['prod_inc', 'sleep_bal', 'new_skill', 'fam_connect', 'relaxed', 'self_time']
    data['lifestyle_change'] = data[change_columns].apply(lambda row: np.any(np.abs(row) > 0.5), axis=1).astype(int)
```

```
[ ] data['lifestyle_change'].value_counts(normalize=True)
```

```
lifestyle_change
1    0.64766
0    0.35234
Name: proportion, dtype: float64
```

```
[ ] from sklearn.preprocessing import StandardScaler
```

```
[ ] numerical_cols = data.select_dtypes(include=['int64', 'float64']).drop(columns='lifestyle_change').columns
    numerical_transformer = StandardScaler()

    # Define preprocessing for categorical columns (encode them)
    categorical_cols = data.select_dtypes(include=['object']).columns
    categorical_transformer = OneHotEncoder(handle_unknown='ignore')
    # Combine preprocessing steps
    preprocessor = ColumnTransformer(
        transformers=[
            ('num', numerical_transformer, numerical_cols),
            ('cat', categorical_transformer, categorical_cols)
        ])

    # Split the target variable from the predictors
    X = data.drop(columns='lifestyle_change')
    y = data['lifestyle_change']

    # Split the data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

    # Preprocess the datasets
    preprocessor.fit(X_train)
    X_train = preprocessor.transform(X_train)
    X_test = preprocessor.transform(X_test)

    X_train.shape, X_test.shape
```

```
((940, 44), (235, 44))
```

```
[ ] from sklearn.preprocessing import OneHotEncoder
    import pickle

    # Fit the OneHotEncoder on your data
    ohe = OneHotEncoder(sparse_output=False)
    ohe.fit(data[['age', 'gender', 'occupation', 'line_of_work', 'prefer', 'certaindays_hw']])

    # Save the OneHotEncoder
    pickle.dump(ohe, open('ohe.pkl', 'wb'))
```

```
[ ] from sklearn.linear_model import LogisticRegression
    from sklearn.metrics import accuracy_score, classification_report
```

```
[ ] model1=LogisticRegression()
```

```
[ ] model1.fit(X_train,y_train)
```

```
▾ LogisticRegression
LogisticRegression()
```

```
[ ] y_pred=model1.predict(X_test)
    y_pred
```

```
array([1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0,
       1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0,
       1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1,
       0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,
       1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1,
       1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1,
       0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1,
       0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1,
       1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1])
```

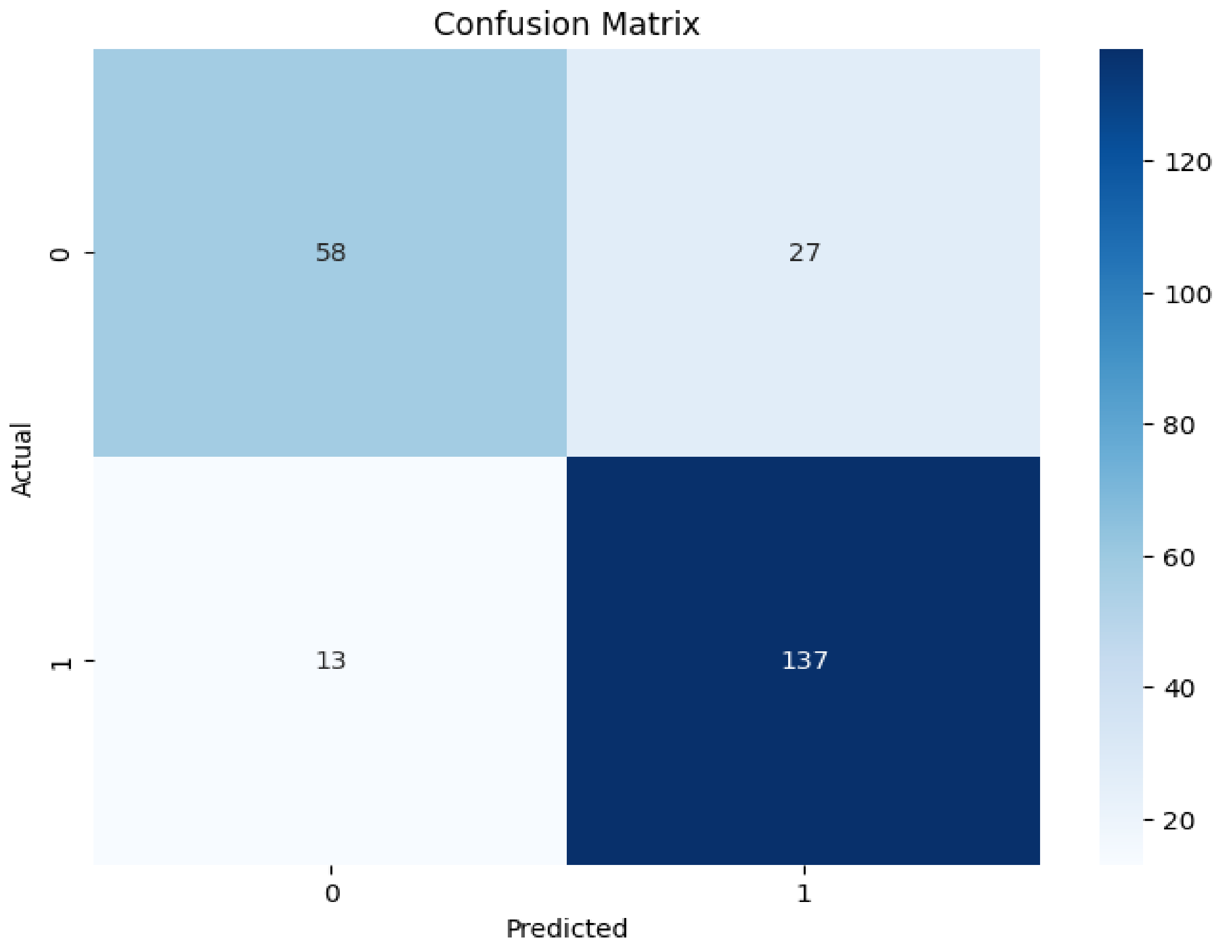
```
[ ] accuracy=accuracy_score(y_test,y_pred)
```

```
[ ] print('Accuracy:',accuracy*100)
    print('\nClassification Report:',classification_report(y_test,y_pred))
```

```
[ ] Accuracy: 82.97872340425532

Classification Report:               precision    recall  f1-score   support

           0       0.82      0.68      0.74        85
           1       0.84      0.91      0.87       150

    accuracy                           0.83       235
   macro avg       0.83      0.80      0.81       235
weighted avg       0.83      0.83      0.83       235
```

```
[ ] from sklearn.metrics import confusion_matrix
```

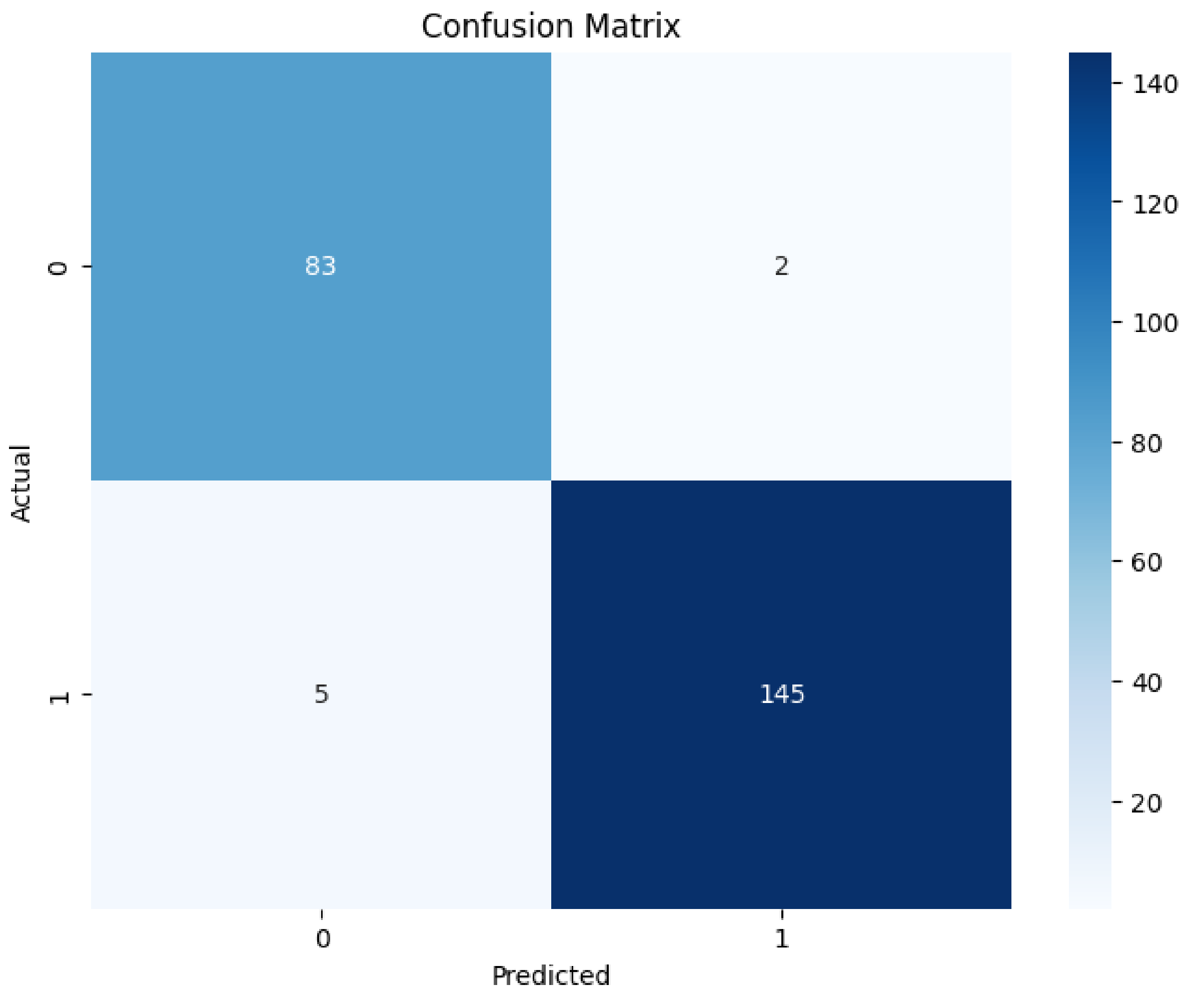
```
[ ] cm = confusion_matrix(y_test, y_pred)

    # Plot the confusion matrix as a heatmap
    plt.figure(figsize=(8, 6))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['0', '1'], yticklabels=['0', '1'])
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    plt.title('Confusion Matrix')
    plt.show()
```




```
[ ] from sklearn.ensemble import RandomForestClassifier
    # Initialize the Random Forest Classifier
    model2 = RandomForestClassifier()

    # Fit the model
    model2.fit(X_train, y_train)

    # Make predictions on the test set
    y_pred = model2.predict(X_test)

    # Model Accuracy
    accuracy = accuracy_score(y_test, y_pred)

    # Evaluate the model
    print("Accuracy: ", accuracy * 100)
    print("\nClassification Report: \n", classification_report(y_test, y_pred))
```

```
Accuracy:  97.02127659574468

Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.98      0.96        85
           1       0.99      0.97      0.98       150

    accuracy                           0.97       235
   macro avg       0.96      0.97      0.97       235
weighted avg       0.97      0.97      0.97       235
```

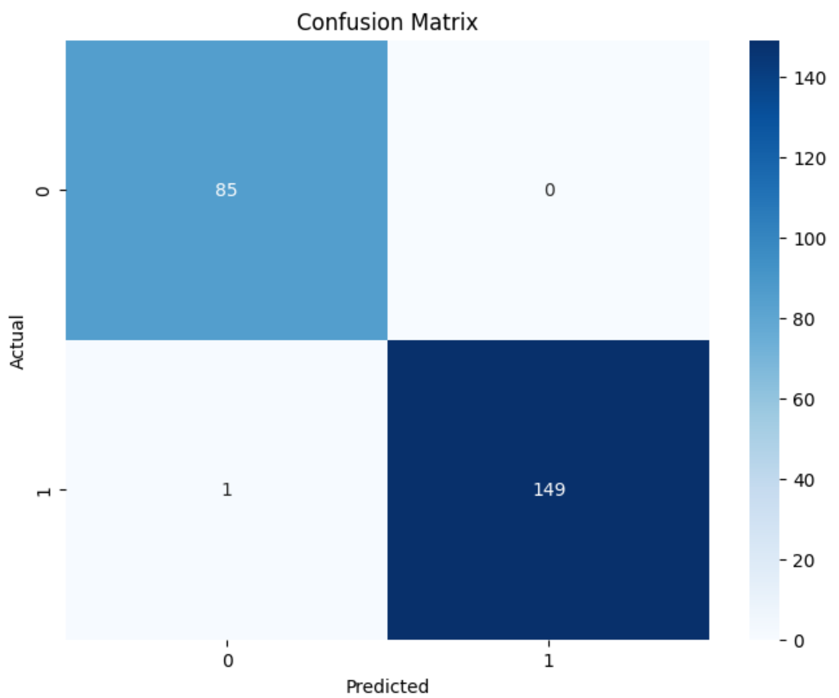
```
cm = confusion_matrix(y_test, y_pred)

# Plot the confusion matrix as a heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['0', '1'], yticklabels=['0', '1'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```



```
[ ] from sklearn.tree import DecisionTreeClassifier

    # Initialize the Decision Tree Classifier
    model3 = DecisionTreeClassifier(random_state=42)

    # Fit the model
    model3.fit(X_train, y_train)

    # Make predictions on the test set
    y_pred = model3.predict(X_test)

    # Model Accuracy
    accuracy = accuracy_score(y_test, y_pred)

    # Evaluate the model
    print("Accuracy: ", accuracy * 100)
    print("\nClassification Report: \n", classification_report(y_test, y_pred))
```

```
Accuracy:  99.57446808510639

Classification Report:
               precision    recall  f1-score   support

           0       0.99      1.00      0.99        85
           1       1.00      0.99      1.00       150

    accuracy                           1.00       235
   macro avg       0.99      1.00      1.00       235
weighted avg       1.00      1.00      1.00       235
```

```
[ ] # Calculate the confusion matrix
    cm = confusion_matrix(y_test, y_pred)

    # Plot the confusion matrix as a heatmap
    plt.figure(figsize=(8, 6))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['0', '1'], yticklabels=['0', '1'])
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    plt.title('Confusion Matrix')
    plt.show()
```



```
[ ] import pickle
```

```
[ ] pickle.dump(model3,open('model.pkl','wb'))
```

```
[ ] pickle.dump(le_age,open('le_age.pkl','wb'))
```

```
[ ] pickle.dump(le_gender,open('le_gender.pkl','wb'))
```

```
[ ] pickle.dump(le_occupation,open('le_occupation.pkl','wb'))
```

```
[ ] pickle.dump(le_line_of_work,open('le_line_of_work.pkl','wb'))
```

```
[ ] pickle.dump(le_prefer,open('le_prefer.pkl','wb'))
```

```
[ ] pickle.dump(le_certaindays_hw,open('le_certaindays_hw.pkl','wb'))
```