EARLY PREDICTION OF LIFESTYLE DISEASES USING MACHINE LEARNING

A PROJECT REPORT

Submitted by,

Mr. Deekshith D L	**- 20201CSD0153**
Mr. Mithun R	**- 20201CSD0182**
Ms. Madhushree A	**- 20201CSD0187**

Under the guidance of,

Mr. LAKSHMISHA S K

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

At

PRESIDENCY UNIVERSITY

BENGALURU

JANUARY 2024

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Project report **"EARLY PREDICTION OF LIFESTYLE DISEASES USING MACHINE LEARNING"** being submitted by **Deekshith D L, Mithun R, Madhushree A** bearing roll number(s) **20201CSD0153, 20201CSD0182, 20201CSD0187** in partial fulfilment of requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering (Data science) is a bonafide work carried out under my supervision.

Mr. LAKSHMISHA S K	**Dr. JAYACHANDRAN ARUMUGAM**
Assistant Professor	Professor & HOD
School of CSE&IS	School of CSE&IS
Presidency University	Presidency University

Dr. C. KALAIARASAN	**Dr. SHAKKEERA L**	**Dr. SAMEERUDDIN KHAN**
Associate Dean	Associate Dean	Dean
School of CSE&IS	School of CSE&IS	School of CSE&IS
Presidency University	Presidency University	Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **EARLY PREDICTION OF LIFESTYLE DISEASES USING MACHINE LEARNING** in partial fulfilment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering (Data Science)**, is a record of our own investigations carried under the guidance of **Mr. LAKSHMISHA S K , ASSISTANT PROFESSOR, School of Computer Science & Engineering , Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

Deekshith D L	**- 20201CSD0153**
Mithun R	**- 20201CSD0182**
Madhushree A	**- 20201CSD0187**

ABSTRACT

Healthcare is a domain where the integration of machine learning (ML) techniques has the potential to revolutionize diagnostics and prognosis. This project presents a comprehensive approach to ML-based disease prediction, focusing on user accessibility and real-time prediction through the deployment of a Flask-based web application.

The project encompasses various stages, beginning with data collection and detailed statistical analysis to understand demographic and vital statistics. A robust data preprocessing pipeline is established to handle missing values and outliers, ensuring the quality of input for subsequent phases.

Feature engineering is critical for extracting relevant information from the dataset, contributing to the creation of an effective ML model. The model development phase involves choosing suitable ML algorithms tailored to the healthcare context. Following this, the models are trained using historical data, and their performance is rigorously evaluated through validation techniques.

Privacy measures are implemented to ensure the secure handling of sensitive health-related data, addressing concerns associated with the deployment of healthcare applications. The deployment of the model into a Flask-based web application enables users to input their health parameters and receive real-time disease predictions.

The prediction results are analyzed, and the models are refined based on the insights gained. Continuous improvement is facilitated through feedback mechanisms and the incorporation of new data, ensuring the adaptability and relevance of the predictive models over time.

This project not only contributes to the growing field of ML in healthcare but also emphasizes the practical implementation of such models through a user-friendly web interface. The Flask integration allows for seamless accessibility, making disease prediction more immediate and actionable for users. The project's success lies in its ability to bridge the gap between advanced ML techniques and practical, user-oriented healthcare applications.

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Dean, School of Computer Science & Engineering , Presidency University and **Dr. Jayachandran Arumugam**, Head of the Department, School of Computer Science & Engineering, Presidency University for getting us permission to undergo the project.

We record our heartfelt gratitude to our beloved Associate Deans

Dr. Kalaiarasan C and Dr. Shakkeera L, School of Computer Science & Engineering, Presidency University for rendering timely help for the successful completion of this project.

We are greatly indebted to our guide **Mr. Lakshmisha S K, Assistant Professor**, School of Computer Science & Engineering , Presidency University for his inspirational guidance, valuable suggestions and providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the University Project-II Coordinators **Dr. Sanjeev P Kaulgud, Dr. Mrutyunjaya MS** and also the department Project Coordinators **Dr. Manjula H M , Mr. Yamanappa**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

Deekshith D L

Mithun R

Madhushree A

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER-1
INTRODUCTION

1.1 General Introduction:

The increasing prevalence of lifestyle diseases has become a significant global health concern, leading to a surge in healthcare costs. Early prediction and preventive measures can play a crucial role in reducing the burden of these diseases. This report focuses on utilizing detailed demographic and vital statistics, obtained during physical checkups, to predict the likelihood of lifestyle diseases. Technology companies, such as Google, can employ machine learning models to enhance early detection, enabling more effective preventive healthcare strategies.

1.2 Introduction to the Domain:

Lifestyle diseases encompass conditions like heart disease, diabetes, hypertension, and obesity, often influenced by factors such as age, genetics, diet, physical activity, and smoking. The objective is to leverage machine learning to analyze these factors and predict the likelihood of individuals developing such diseases, facilitating timely interventions.

1.3 Project Overview:

The ML Disease Prediction Web Application is a comprehensive solution designed to leverage machine learning techniques for predicting various health conditions based on user-provided information. The central component of the project is a user-friendly web application developed using Flask, allowing users to input their health-related details and receive real-time predictions about potential diseases.

CHAPTER-2
LITERATURE REVIEW

2.1 Introduction

The intersection of machine learning and healthcare has garnered significant attention in recent years, with researchers exploring predictive modeling to identify health conditions based on lifestyle factors. This literature review aims to provide insights into existing models and methodologies used in projects similar to ours, which focuces on predicting diseases related to lifestyle choices.

2.2 Papers:

Sl.No.	Name Of Author's	Paper Title	Technique
1.	A. Parab, P. Gholap and V. Patankar	"DiseaseLens: A Lifestyle related Disease Predictor," *2022 5th International Conference on Advances in Science and Technology (ICAST), 2022 5th International Conference on Advances in Science and Technology (ICAST),*	Decision Tree
2.	Xia Yu and Shuoyu Wang	"A Health Check and Prediction System for Lifestyle-Related Disease Prevention," *First International Conference on Innovative Computing, Information and Control - Volume I (ICICIC'06)*	Fuzzy sets
3.	M. Govindaraj, V. Asha, B. Saju, M. Sagar and Rahul	"Machine Learning Algorithms for Disease Prediction Analysis," *2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)*	Data mining,Machine learning techniques,87% accurate

Advantages:	Limitations:
● **Early Detection:** Existing methods leverage medical data for early disease detection. ● **Cost Reduction:** Early identification enables cost-effective preventive measures. ● **Improved Patient Outcomes:** Timely interventions lead to better patient outcomes.	● **Limited Predictive Accuracy:** Current methods may have limitations inaccurately predicting disease risk. ● **Dependency on Data Quality:** The reliability of predictions depends on the quality of input data. ● **Ethical Considerations:** Ethical concerns arise from the use of personal health information.

CHAPTER-3
PROPOSED METHODOLOGY

Hardware and Software Used:

Hardware:

- High-performance computing system for model training and validation.

Software:

- Python for programming.
- Scikit-learn for machine learning model development.
- Jupyter Notebooks for experimentation and analysis.

Methodology:

A. Data Collection:

The project utilized a carefully dataset encompassing crucial health related information such as age, gender, BMI, smoking status, blood pressure, physical activity, diet, cholesterol levels, alcohol consumption, and sleep duration. The dataset underwent preprocessing to handle missing values and encode categorical variables appropriately.

B. Machine Learning Model:

A Random Forest Classifier was chosen for its ability to handle complex relationships in the data. The model underwent training using the preprocessed dataset, with an emphasis on hyperparameter tuning to enhance its predictive performance.

C. Web Application Development:

Flask, a lightweight Python web framework, was employed to build the user interface. HTML forms were designed to capture user input, which was then seamlessly processed and sent to the machine learning model for real-time predictions. The application's visual elements were styled using HTML and CSS to enhance the overall user experience.

CHAPTER-4
OBJECTIVES

General:

- Develop a machine learning model for predicting lifestyle diseases.
- Enhance predictive accuracy by incorporating advanced features.
- Address privacy concerns through secure data handling practices.
- Evaluate the model's effectiveness in a real-world setting.

Aim:

1. **Machine Learning Model:**
   Develop a robust machine learning model capable of accurately predicting diseases based on diverse input features.
2. **Web Application Development:**
   Implement a user-friendly web application using Flask, providing an interactive and seamless interface for users to input health details and receive instantaneous disease predictions.
3. **Real-time Recommendations:**
   Incorporate features that provide users with practical recommendations and precautions based on the predicted diseases to promote a healthier lifestyle.

CHAPTER-5
SYSTEM DESIGN

Procedure:

- **Data Collection:** Gather detailed demographic and vital stats from individuals during physical checkups.
- **Data Preprocessing:** Clean and preprocess the data, handling missing values and outliers.
- **Feature Engineering:** Extract relevant features, considering factors like age, gender, BMI, smoking status, etc.
- **Model Development:** Implement machine learning models such as random forest classifier, decision trees, or neural networks.
- **Model Training:** Train the models using historical data, fine-tuning parameters for optimal performance.
- **Validation:** Evaluate the model's performance using a separate dataset.
- **Privacy Measures:** Implement encryption and secure data handling practices to address privacy concerns.
- **Deployment:** Deploy the model for real-world prediction during physical checkups.

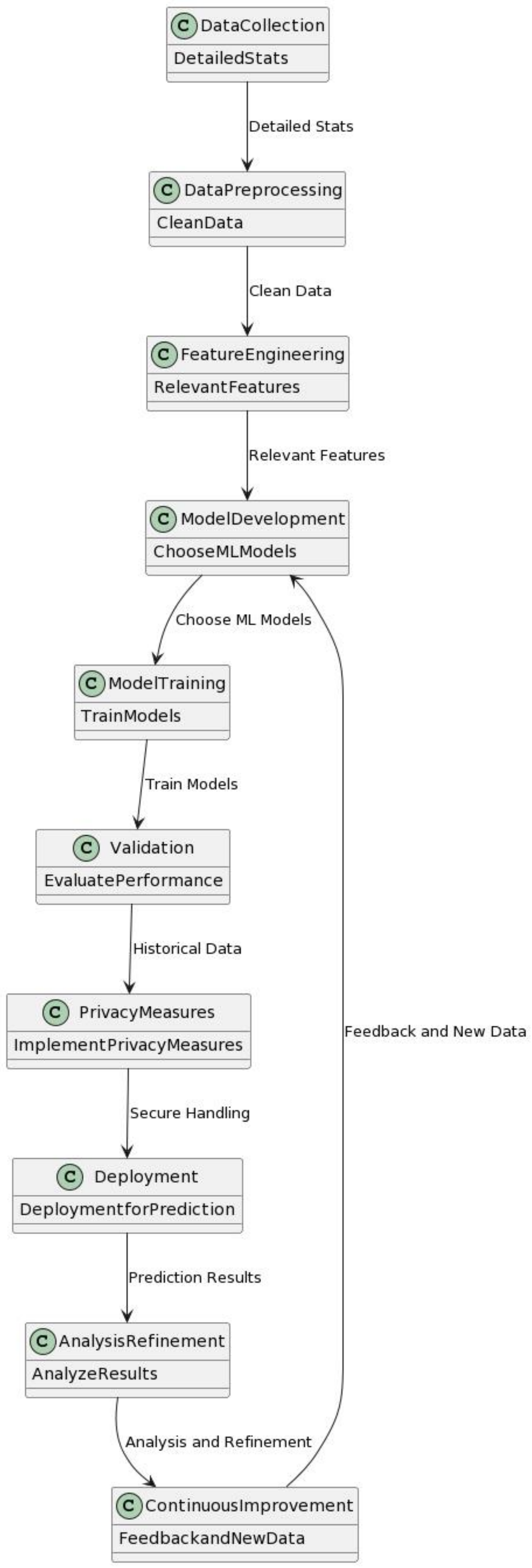

DataCollection
DetailedStats
Detailed Stats
DataPreprocessing
CleanData
Clean Data
FeatureEngineering
RelevantFeatures
Relevant Features
ModelDevelopment
ChooseMLModels
Choose ML Models
ModelTraining
TrainModels
Train Models
Validation
EvaluatePerformance
Historical Data
PrivacyMeasures
ImplementPrivacyMeasures
Secure Handling
Deployment
DeploymentforPrediction
Prediction Results
AnalysisRefinement
AnalyzeResults
Analysis and Refinement
ContinuousImprovement
FeedbackandNewData
Feedback and New Data

Figure 5.1

CHAPTER-6
IMPLEMENTATION

Code Structure:

- **Early_Prediction_of_Lifestyle_diseases.ipynb**

  A. **Data Collection**
  B. **Data Preprocessing**
  C. **Feature Engineering:**
  D. **Model Development:**
  E. **Model Training**
  F. Validation

- **app.py:**

  Main Flask application handling web routes and model predictions.

- **model_preprocessor.pkl**
- **templates/index.html**

  HTML template for user input page.

- **templates/result.html**

  HTML template for displaying predictionst.

- **templates/style.css**

  CSS file for styling.

A. Data Preprocessing:

- **Handling Missing Values:** Robust techniques were employed to address missing values and maintain the integrity of the dataset.
- **Categorical Variable Encoding:** Necessary encoding of categorical variables was performed to prepare the data for machine learning.

B. Machine Learning Model:

- **Random Forest Classifier:** Implementation of the Random Forest Classifier using the scikit-learn library.
- **Model Training:** The model was trained on the preprocessed dataset, considering diverse health parameters.
- **Evaluation:** Rigorous evaluation of the model's performance on a separate testing set to ensure predictive accuracy.

C. Web Application:

- **HTML Templates:** Creation of HTML templates for capturing user input and displaying predictions.
- **Flask Integration:** Development of a Flask web application to seamlessly handle user requests, interface with the machine learning model, and present real-time predictions.

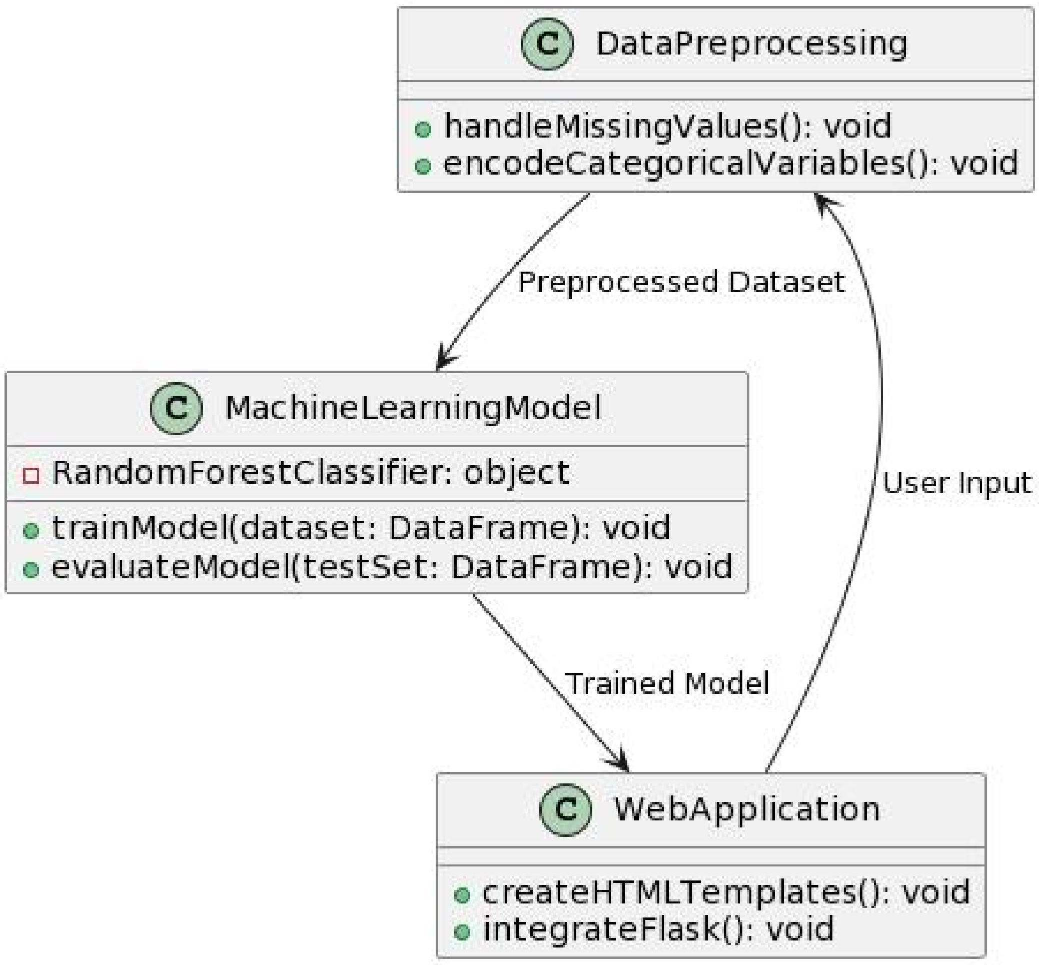
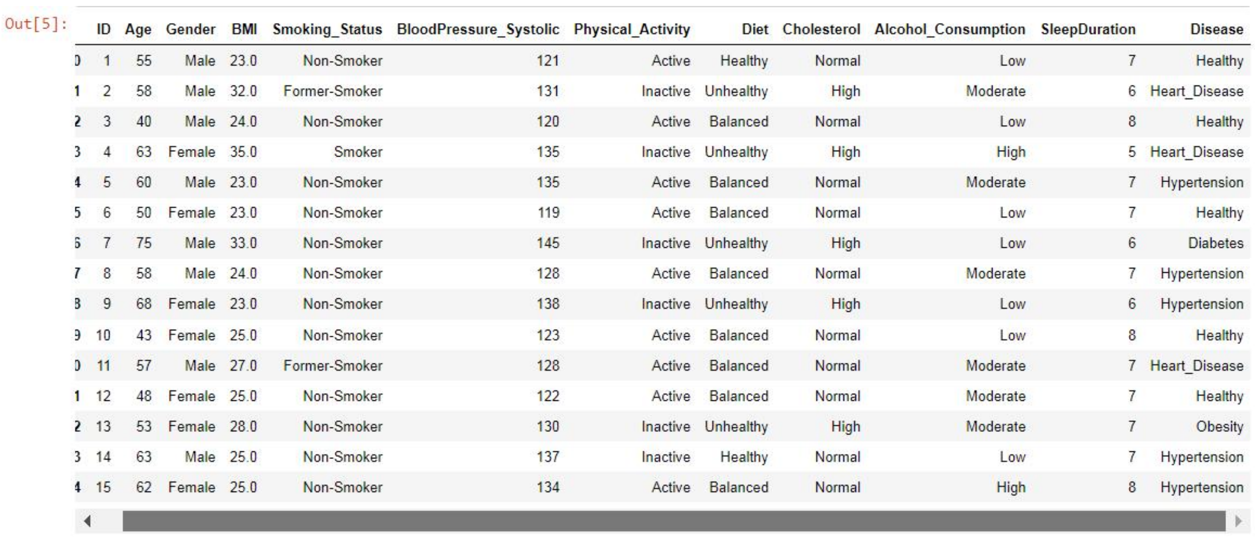

DataPreprocessing
+ handleMissingValues(): void
+ encodeCategoricalVariables(): void
Preprocessed Dataset
MachineLearningModel
- RandomForestClassifier: object
+ trainModel(dataset: DataFrame): void
+ evaluateModel(testSet: DataFrame): void
User Input
Trained Model
WebApplication
+ createHTMLTemplates(): void
+ integrateFlask(): void

Figure 6.1 Implementation block diagram

1. Data collection:



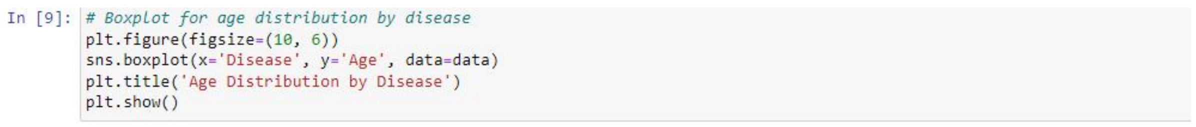
```
Out[5]:
     ID  Age  Gender   BMI  Smoking_Status  BloodPressure_Systolic  Physical_Activity       Diet  Cholesterol  Alcohol_Consumption  SleepDuration        Disease
0     1   55    Male  23.0      Non-Smoker                     121             Active    Healthy       Normal                  Low              7        Healthy
1     2   58    Male  32.0   Former-Smoker                     131           Inactive  Unhealthy         High             Moderate              6  Heart_Disease
2     3   40    Male  24.0      Non-Smoker                     120             Active   Balanced       Normal                  Low              8        Healthy
3     4   63  Female  35.0          Smoker                     135           Inactive  Unhealthy         High                 High              5  Heart_Disease
4     5   60    Male  23.0      Non-Smoker                     135             Active   Balanced       Normal             Moderate              7   Hypertension
5     6   50  Female  23.0      Non-Smoker                     119             Active   Balanced       Normal                  Low              7        Healthy
6     7   75    Male  33.0      Non-Smoker                     145           Inactive  Unhealthy         High                  Low              6       Diabetes
7     8   58    Male  24.0      Non-Smoker                     128             Active   Balanced       Normal             Moderate              7   Hypertension
8     9   68  Female  23.0      Non-Smoker                     138           Inactive  Unhealthy         High                  Low              6   Hypertension
9    10   43  Female  25.0      Non-Smoker                     123             Active   Balanced       Normal                  Low              8        Healthy
0    11   57    Male  27.0   Former-Smoker                     128             Active   Balanced       Normal             Moderate              7  Heart_Disease
1    12   48  Female  25.0      Non-Smoker                     122             Active   Balanced       Normal             Moderate              7        Healthy
2    13   53  Female  28.0      Non-Smoker                     130           Inactive  Unhealthy         High             Moderate              7        Obesity
3    14   63    Male  25.0      Non-Smoker                     137           Inactive    Healthy       Normal                  Low              7   Hypertension
4    15   62  Female  25.0      Non-Smoker                     134             Active   Balanced       Normal                 High              8   Hypertension
```

Attributes:

- ID - Identification Number
- Age - Age of a Person
- BMI - Body Mass Index of a Person
- Smoking_Systolic - It is a Blood Pressure range
- Physical Activity - Person Daily Activity Based(Active, Inactive)
- Diet - Person Diet (Healthy, Unhealthy, balanced)
- Cholesterol - Cholesterol level of person (Normal, High)
- Alcohol_Consumption - Low, High, Moderate.
- Sleep_Duration - Range from sleep cycle(5-9)
- Disease - Healthy, Hypertension, Heart_Disease, Obesity, Diabetes.(Output feature)

2. Exploratory Data Analysis:

[i] Box plot



```
In [9]: # Boxplot for age distribution by disease
        plt.figure(figsize=(10, 6))
        sns.boxplot(x='Disease', y='Age', data=data)
        plt.title('Age Distribution by Disease')
        plt.show()
```

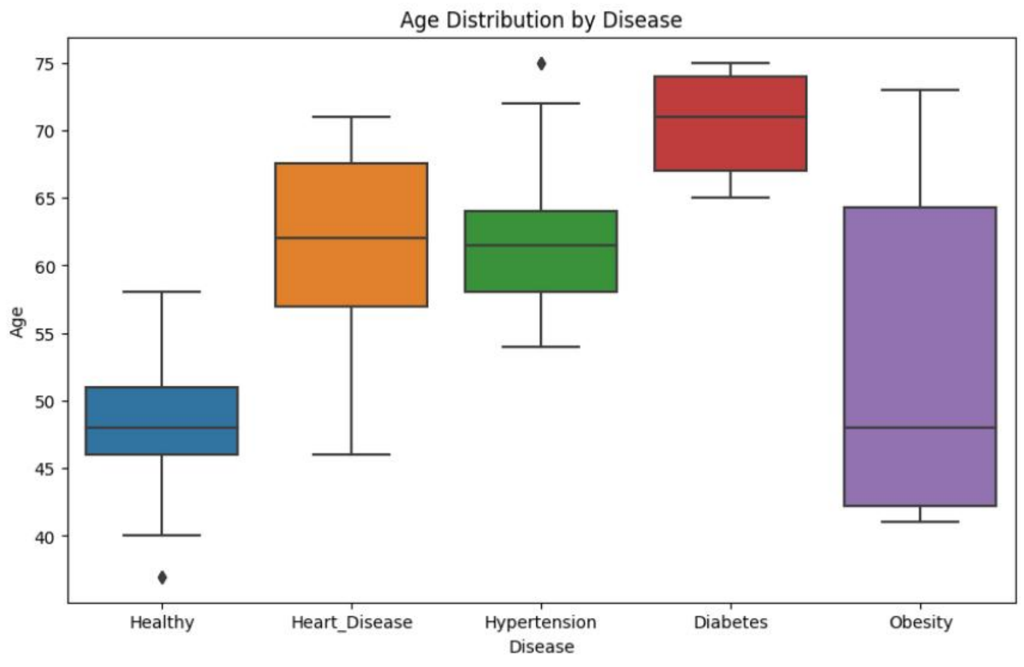
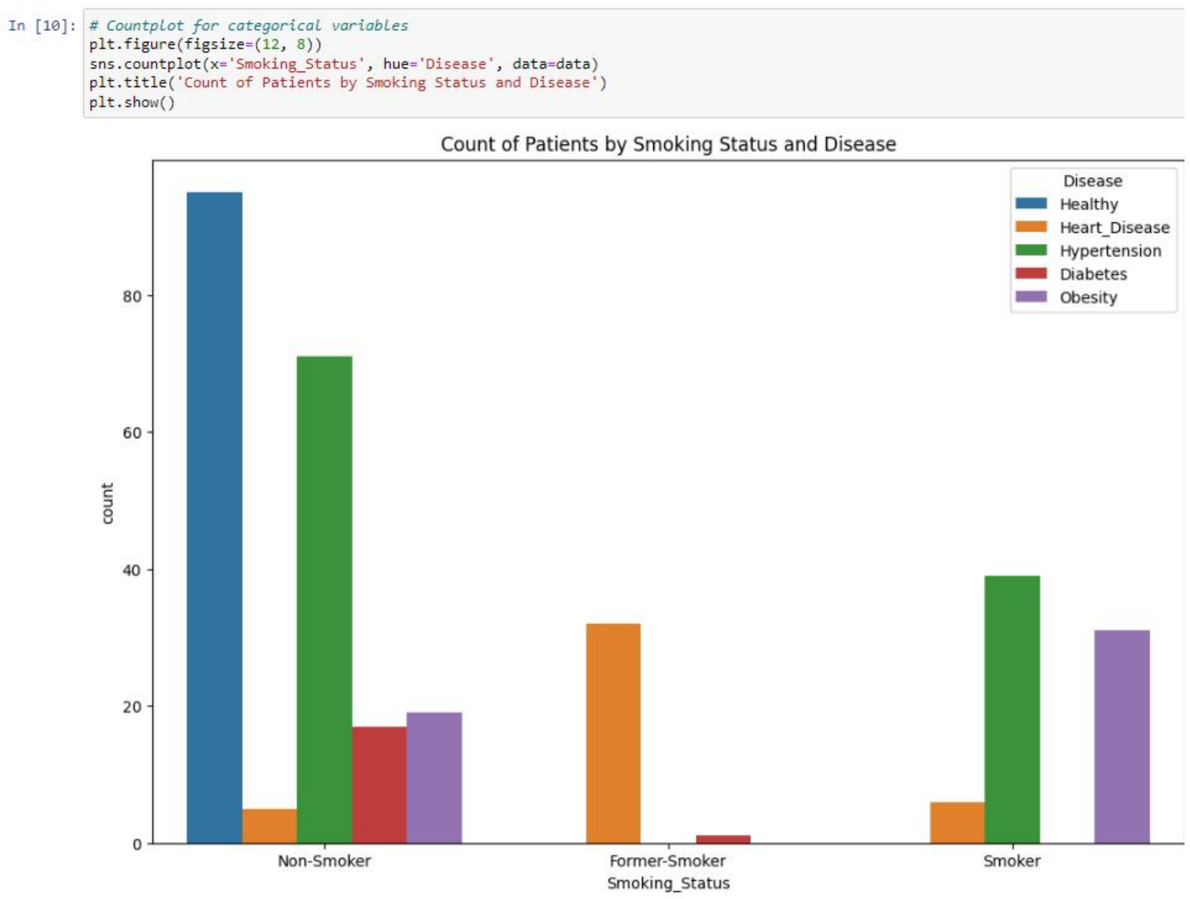


Figure 6.2 Age v/s Disease Box plot

Box plots offer a concise visual representation of a dataset's value variability. The dataset's minimum and maximum values, the upper and lower quartile , the median, and any outliers are displayed. Outliers in data might highlight errors or oddities.

It shows the Mean , Median range of the Age over the Early lifestyle Diseases.

[ii] Count Plot



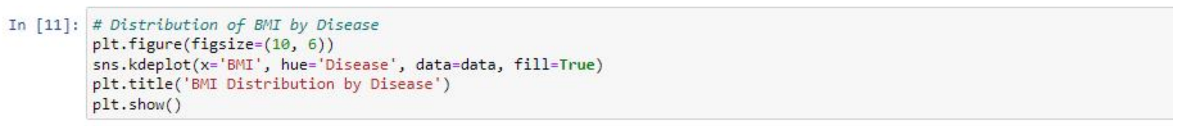
```
In [10]: # Countplot for categorical variables
         plt.figure(figsize=(12, 8))
         sns.countplot(x='Smoking_Status', hue='Disease', data=data)
         plt.title('Count of Patients by Smoking Status and Disease')
         plt.show()
```

Figure 6.3 Countplot over the Smoking Status and Diseases.

Here We count the count plot against the Smoking Stats among the Diseases.

Count Plot is utilised to show the number of categorical observations in each data set bin.

[iii] KDE Plot



```
In [11]: # Distribution of BMI by Disease
         plt.figure(figsize=(10, 6))
         sns.kdeplot(x='BMI', hue='Disease', data=data, fill=True)
         plt.title('BMI Distribution by Disease')
         plt.show()
```

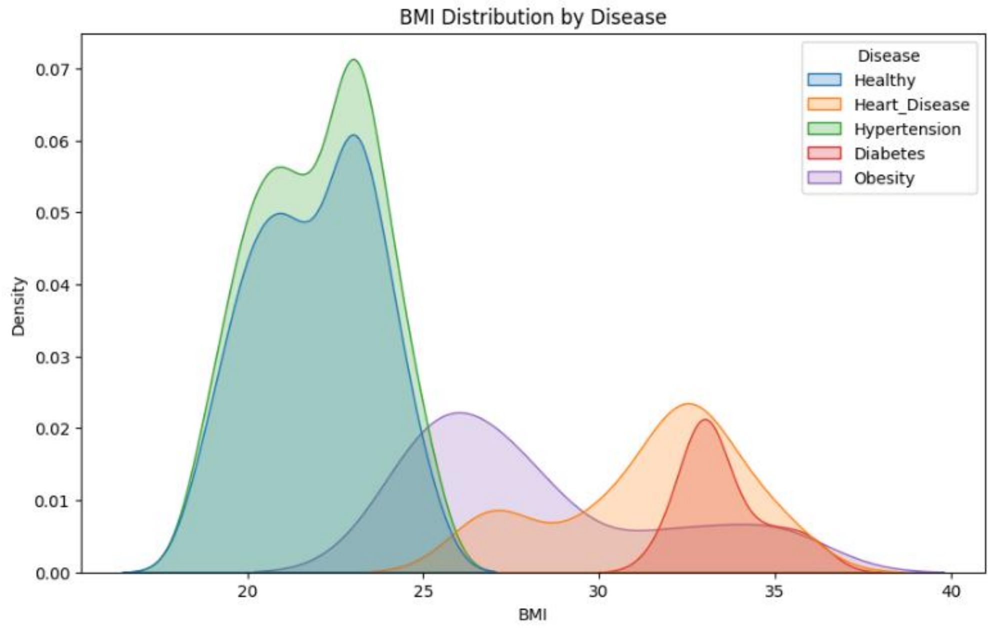


Figure 6.4 BMI Distribution By Diseases.

The KDE plot offers information on the shape, central tendency, and distribution of data by estimating the probability density function of a continuous variable.

Here the Distribution among the BMI and Density Hypertension is the highest density, then Healthy.

3. Data Preprocessing:

Data preprocessing is a crucial step in preparing datasets for machine learning models. It involves handling missing values, transforming categorical variables into a format suitable for modeling, and scaling or normalizing numerical features.

1. Identifying Categorical and Numerical Columns:

```
    categorical_cols = ['Gender', 'Smoking_Status', 'Diet', 'Alcohol_Consumption',
'Physical_Activity', 'Cholesterol']
    numerical_cols = ['Age', 'BMI', 'BloodPressure_Systolic', 'SleepDuration']
```

2. Creating Transformers for Numerical and Categorical Columns.
   - Numeric Transformer:
     - It uses the SimpleImputer class to fill missing numerical values with the mean of the column.

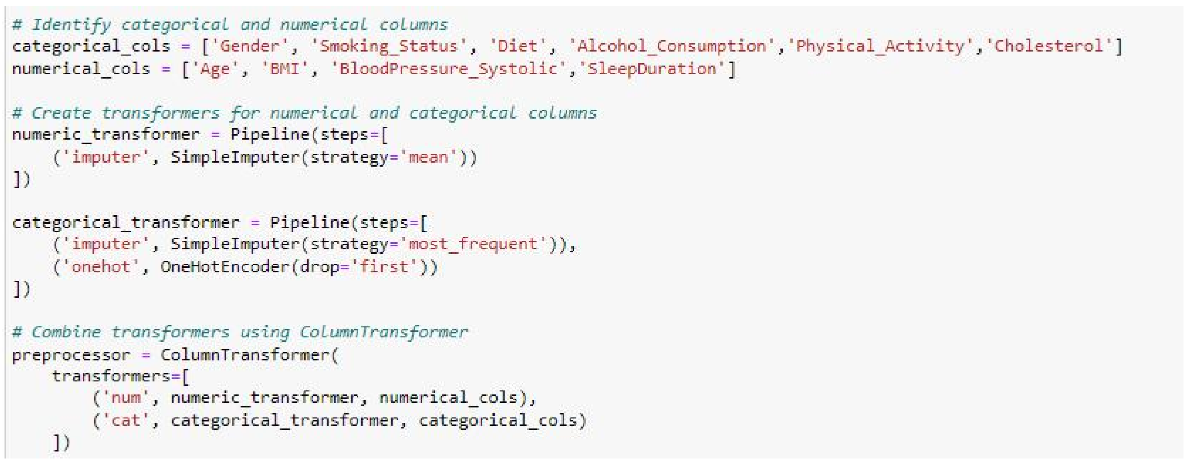
   - Categorical Transformer:
     - It uses the SimpleImputer class to fill missing categorical values with the most frequent value in the column.
     - It then applies OneHotEncoder to convert categorical variables into a one-hot encoded format, dropping the first column to avoid multicollinearity.

3. Combine Transformers using ColumnTransformer:

```
    preprocessor = ColumnTransformer(
        transformers=[
            ('num', numeric_transformer, numerical_cols),
            ('cat', categorical_transformer, categorical_cols)
        ])
```

The `ColumnTransformer` combines the transformers defined for numerical and categorical columns. It applies the specified transformations to the respective columns, resulting in a preprocessed dataset that can be used for training machine learning models.

This preprocessed data can then be used for training and evaluating machine learning models. The preprocessing steps encapsulated in the transformers help ensure that the data is in a suitable format for various algorithms.



```python
# Identify categorical and numerical columns
categorical_cols = ['Gender', 'Smoking_Status', 'Diet', 'Alcohol_Consumption','Physical_Activity','Cholesterol']
numerical_cols = ['Age', 'BMI', 'BloodPressure_Systolic','SleepDuration']

# Create transformers for numerical and categorical columns
numeric_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='mean'))
])

categorical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='most_frequent')),
    ('onehot', OneHotEncoder(drop='first'))
])

# Combine transformers using ColumnTransformer
preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, numerical_cols),
        ('cat', categorical_transformer, categorical_cols)
    ])
```

Figure 6.5 Data Preprocessing

4. Model Building:

1. Splitting the Dataset:
   - `X` contains the features (independent variables) of the dataset.
   - `y` contains the target variable (dependent variable), which is 'Disease' in this case.

2. Splitting into Training and Testing Sets:
   - The dataset is divided into training and testing sets using train_test_split.
   - 80% of the data is used for training (X_train and y_train), and 20% is reserved for testing (`X_test and  y_test`).

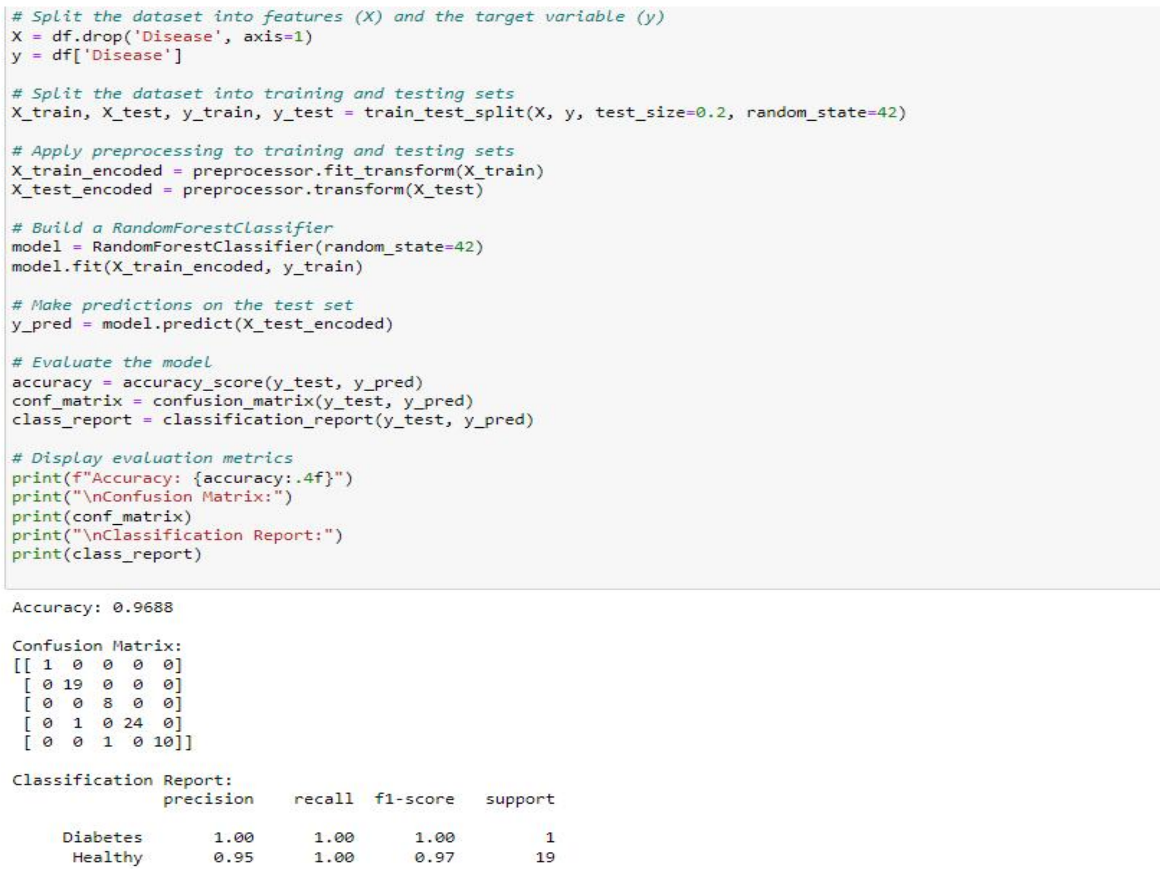
3. Applying Preprocessing:
   - The previously defined `preprocessor` (from your earlier code) is applied to both the training and testing sets to transform and preprocess the data.

4. Building a RandomForestClassifier:
   - A RandomForestClassifier is instantiated and trained using the preprocessed training data (X_train_encoded and y_train).

5. Making Predictions:
   - The trained model is used to make predictions on the preprocessed testing data.

6. Evaluating the Model:
   - The accuracy, confusion matrix, and classification report are computed to evaluate the model's performance on the test set.

7. Displaying Evaluation Metrics:
   - The accuracy, confusion matrix, and classification report are printed to the console to provide a comprehensive overview of the model's performance.



```
# Split the dataset into features (X) and the target variable (y)
X = df.drop('Disease', axis=1)
y = df['Disease']

# 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)

# Apply preprocessing to training and testing sets
X_train_encoded = preprocessor.fit_transform(X_train)
X_test_encoded = preprocessor.transform(X_test)

# Build a RandomForestClassifier
model = RandomForestClassifier(random_state=42)
model.fit(X_train_encoded, y_train)

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

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

# Display evaluation metrics
print(f"Accuracy: {accuracy:.4f}")
print("\nConfusion Matrix:")
print(conf_matrix)
print("\nClassification Report:")
print(class_report)
```

```
Accuracy: 0.9688

Confusion Matrix:
[[ 1  0  0  0  0]
 [ 0 19  0  0  0]
 [ 0  0  8  0  0]
 [ 0  1  0 24  0]
 [ 0  0  1  0 10]]

Classification Report:
              precision    recall  f1-score   support

    Diabetes       1.00      1.00      1.00         1
     Healthy       0.95      1.00      0.97        19
```

Figure 6.6 Model Building

Accuracy: 0.9688

- The accuracy of the model is 96.88%, indicating the proportion of correctly classified instances in the test set.

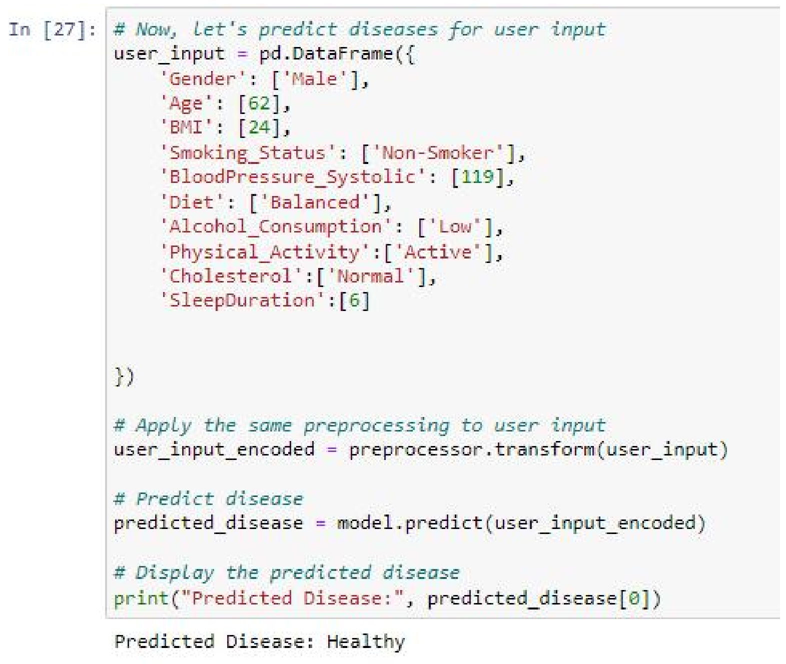
Confusion Matrix:

- The confusion matrix provides a breakdown of the model's predictions compared to the actual class labels.
- Each row represents the actual class, and each column represents the predicted class.

Classification Report:

- The precision, recall, and F1-score are provided for each class, along with the support (number of instances) for each class.
- Precision is the ratio of correctly predicted positive observations to the total predicted positives. High precision indicates low false positive rate.
- Recall (Sensitivity or True Positive Rate) is the ratio of correctly predicted positive observations to the total actual positives. High recall indicates low false negative rate.
- F1-score is the weighted average of precision and recall, providing a balance between the two metrics.

5. Predictive System:



```
In [27]: # Now, let's predict diseases for user input
         user_input = pd.DataFrame({
             'Gender': ['Male'],
             'Age': [62],
             'BMI': [24],
             'Smoking_Status': ['Non-Smoker'],
             'BloodPressure_Systolic': [119],
             'Diet': ['Balanced'],
             'Alcohol_Consumption': ['Low'],
             'Physical_Activity':['Active'],
             'Cholesterol':['Normal'],
             'SleepDuration':[6]


         })

         # Apply the same preprocessing to user input
         user_input_encoded = preprocessor.transform(user_input)

         # Predict disease
         predicted_disease = model.predict(user_input_encoded)

         # Display the predicted disease
         print("Predicted Disease:", predicted_disease[0])

         Predicted Disease: Healthy
```

Figure 6.7 Predictive system

1. User Input:

- A sample user input is created as a Pandas DataFrame, containing various health-related features such as gender, age, BMI, smoking status, blood pressure, diet, alcohol consumption, physical activity, cholesterol level, and sleep duration.

2. Preprocessing User Input:

- The same preprocessing steps that were applied during the training and testing phases are utilized to transform the user input into a format compatible with the model's expectations. This ensures consistency and compatibility between the training data and the user input.

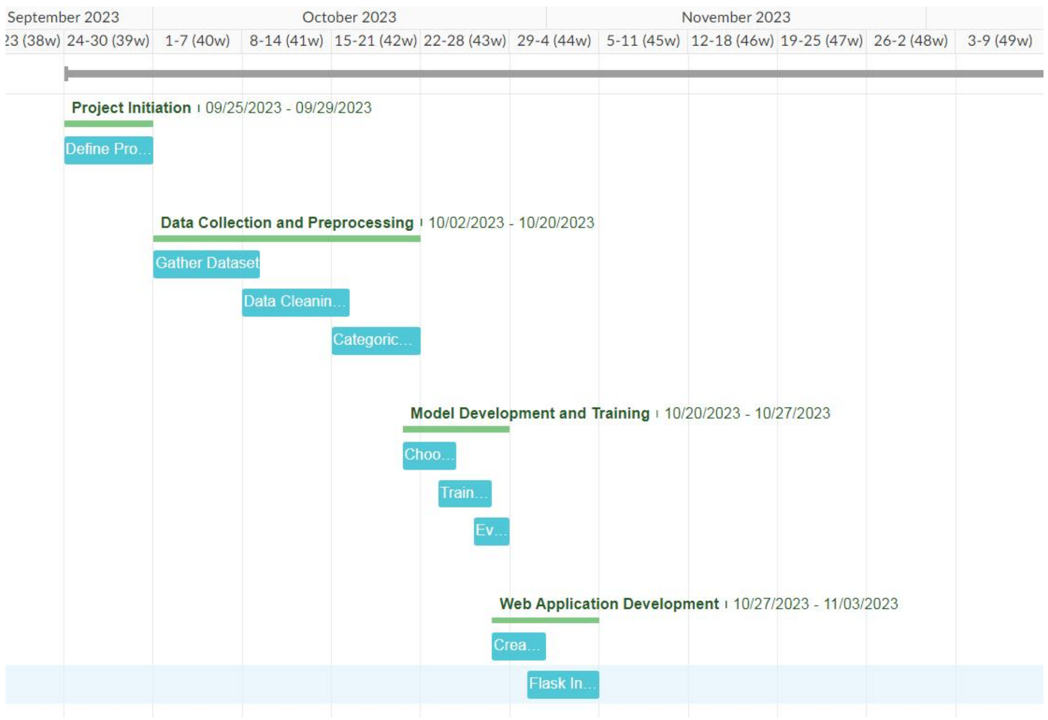
3. Predicting Disease:

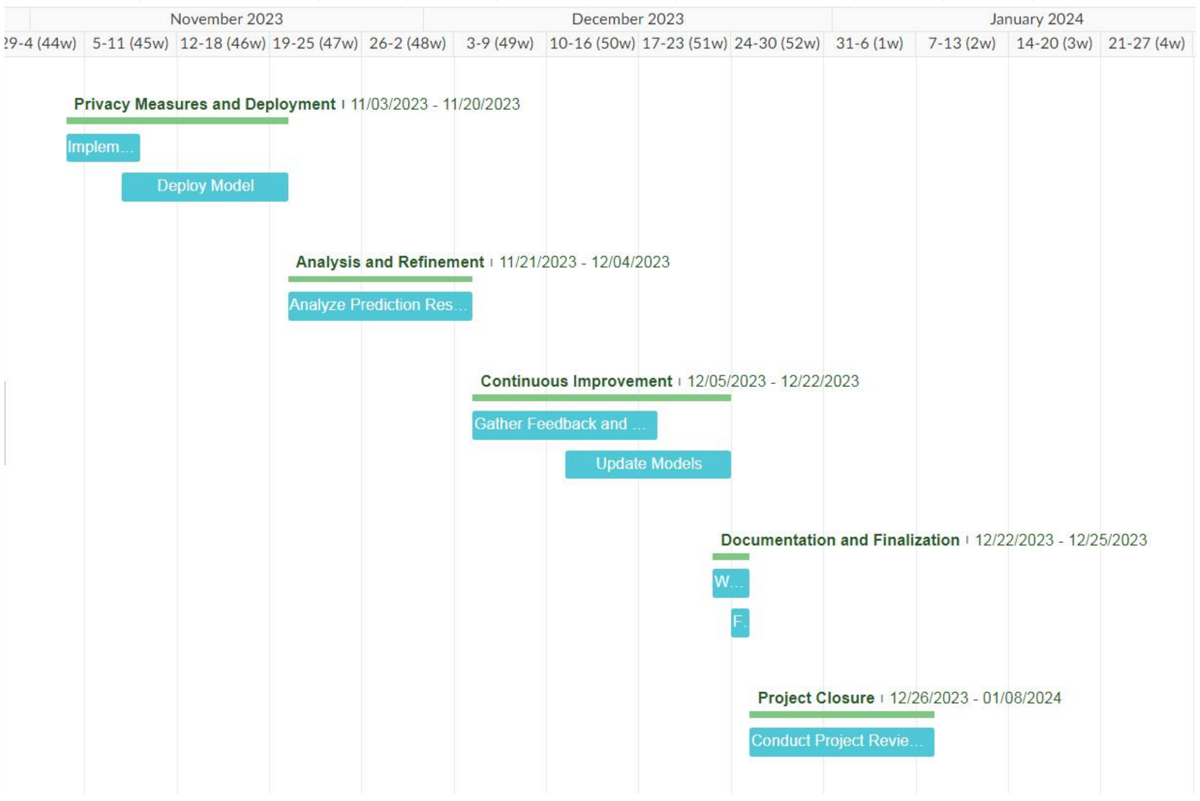
- The preprocessed user input is fed into the pre-trained machine learning model (`model`) to predict the likely disease based on the provided health parameters.

4. Displaying Predicted Disease:

- The predicted disease is printed to the console using the `print` statement.

CHAPTER-7
TIMELINE FOR EXECUTION OF PROJECT
(GANTT CHART)





1. Project Initiation

2. Data Collection and preprocessing

3. Model Development and Training

4. Web Application Development

5. Deployment

6. Analysis and Refinement

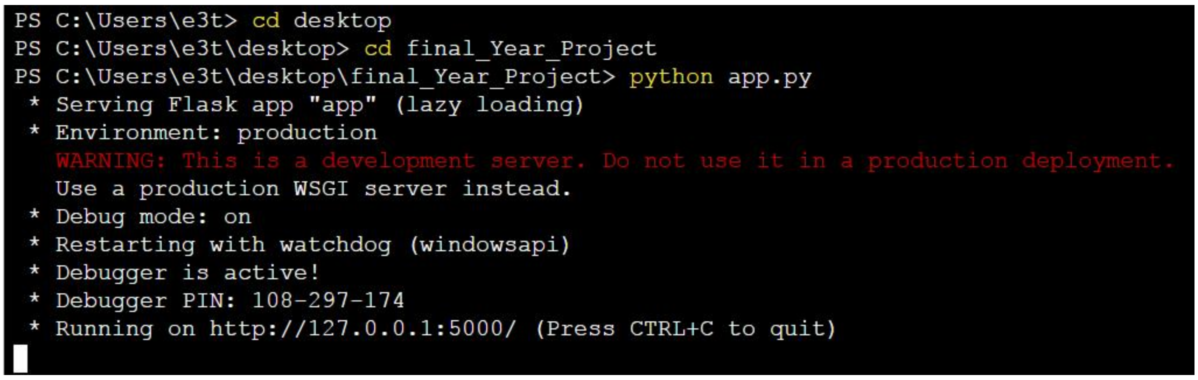
7. Continues Improvement

8. Documentation And finalization

9. Project Closure

CHAPTER-8
OUTCOMES

Execution procedure:



```
PS C:\Users\e3t> cd desktop
PS C:\Users\e3t\desktop> cd final_Year_Project
PS C:\Users\e3t\desktop\final_Year_Project> python app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 108-297-174
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Figure 8.1 Execution procedure

Above is the Execution Procedure:

We can execute the project in Command Prompt terminal.

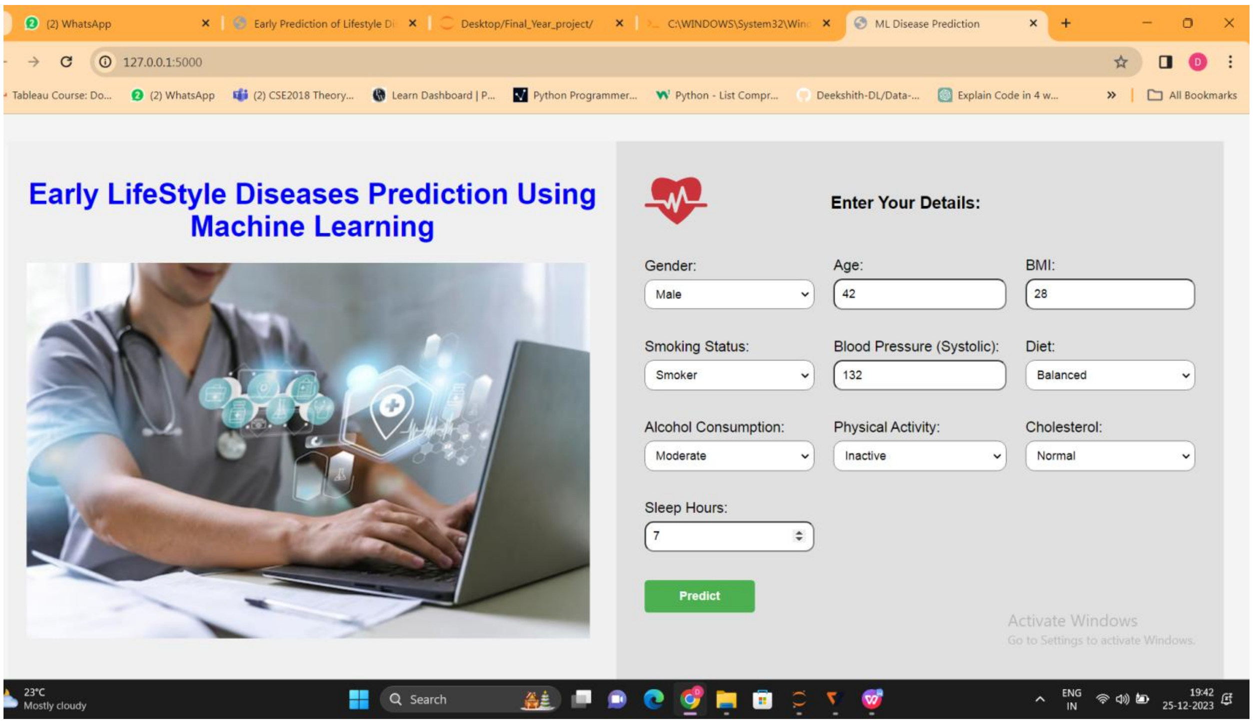
First we redirect to the Project Folder.

Then in The Folder We Start Executing the main app.py file with integration of Machine Learning model and User Interface pages.

Excecute with the command python app.py.

Then the resulted page redirects to the Local Host page.

Main Page:



Early LifeStyle Diseases Prediction Using Machine Learning

Enter Your Details:

Gender: Male

Age: 42

BMI: 28

Smoking Status: Smoker

Blood Pressure (Systolic): 132

Diet: Balanced

Alcohol Consumption: Moderate

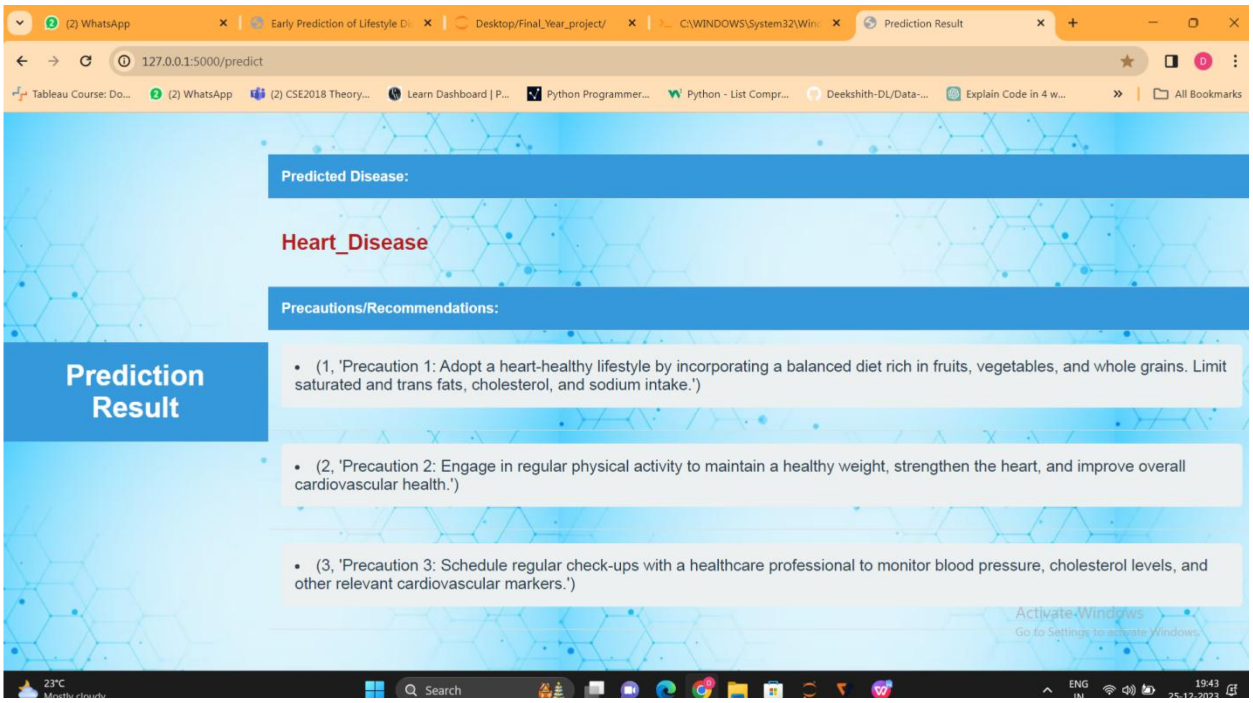
Physical Activity: Inactive

Cholesterol: Normal

Sleep Hours: 7

Figure 8.2 Main Page of User Input

Predicted Result:



Predicted Disease:

Heart_Disease

Precautions/Recommendations:

- (1, 'Precaution 1: Adopt a heart-healthy lifestyle by incorporating a balanced diet rich in fruits, vegetables, and whole grains. Limit saturated and trans fats, cholesterol, and sodium intake.')
- (2, 'Precaution 2: Engage in regular physical activity to maintain a healthy weight, strengthen the heart, and improve overall cardiovascular health.')
- (3, 'Precaution 3: Schedule regular check-ups with a healthcare professional to monitor blood pressure, cholesterol levels, and other relevant cardiovascular markers.')

Figure 8.3 Result Page for the User

Heart_Disease:

- "Precaution 1: Adopt a heart-healthy lifestyle by incorporating a balanced diet rich in fruits, vegetables, and whole grains. Limit saturated and trans fats, cholesterol, and sodium intake."
- "Precaution 2: Engage in regular physical activity to maintain a healthy weight, strengthen the heart, and improve overall cardiovascular health."
- "Precaution 3: Schedule regular check-ups with a healthcare professional to monitor blood pressure, cholesterol levels, and other relevant cardiovascular markers."

Obesity:

- "Precaution 1: Establish and maintain a well-balanced diet that includes appropriate portion sizes and avoids excessive consumption of high-calorie, low-nutrient foods."
- "Precaution 2: Engage in regular physical activity, combining both aerobic exercises and strength training, to support weight management.",
- "Precaution 3: Seek guidance from healthcare professionals or nutritionists for personalized weight loss strategies and lifestyle modifications."

Diabetes:

- "Precaution 1: Manage and maintain a healthy weight through a combination of a well-balanced diet and regular physical activity.",
- "Precaution 2: Monitor blood glucose levels as recommended by healthcare professionals and adhere to prescribed medications or insulin regimens.",
- "Precaution 3: Educate oneself about diabetes management, including understanding the impact of diet, exercise, and stress on blood sugar levels."

Hypertension:

- "Precaution 1: Adopt a low-sodium diet by reducing the intake of processed foods and incorporating more fresh fruits, vegetables, and lean proteins.",
- "Precaution 2: Engage in regular aerobic exercise, such as brisk walking, swimming, or cycling, to help control blood pressure",
- "Precaution 3: Limit alcohol consumption and avoid tobacco products, as they can contribute to elevated blood pressure."

Healthy:

- "Recommendation 1: Continue maintaining a healthy lifestyle with a balanced diet, regular exercise, and proper stress management.",
- "Recommendation 2: Schedule routine health check-ups to monitor overall well-being and catch any potential health issues early.",
- "Recommendation 3: Stay informed about preventive healthcare measures and consider health screenings appropriate for one's age and risk factors."

CHAPTER-9
RESULTS AND DISCUSSIONS

Result:

The model demonstrated accurate predictions on the test dataset, achieving a high level of precision across multiple diseases. The web application successfully integrates with the trained model, providing users with instant predictions based on their input.

The machine learning model exhibited commendable performance during evaluation, showcasing its efficacy in predicting diseases accurately. The web application successfully facilitated user interactions, capturing health-related details and delivering predictions in a user-friendly manner.

Discussion:

The project discusses the significance of integrating machine learning models into web applications for health-related predictions. The application can be a valuable tool for individuals to assess their health risks and make informed decisions.

Future Work:

Future work includes expanding the dataset, incorporating additional features, and refining the model for enhanced prediction accuracy. User feedback
will be collected to improve the web application's usability.

CHAPTER-10
CONCLUSION

- This project aims to significantly contribute to preventive healthcare by leveraging machine learning for early prediction of lifestyle diseases.

- Through comprehensive data analysis, the model is expected to provide accurate predictions, allowing for timely interventions and cost-effective healthcare.

- The ML Disease Prediction Web Application provides users with a valuable tool to assess potential health risks based on lifestyle and health-related information. The integration of a powerful machine learning model with an intuitive web interface contributes to a user-friendly and informative health assessment experience.

- In conclusion, the project successfully demonstrates the feasibility of developing a web application for disease prediction using machine learning. The integration of a predictive model into a user-friendly interface opens avenues for preventive healthcare and early intervention.

REFERENCES

[1] A. Parab, P. Gholap and V. Patankar , "DiseaseLens: A Lifestyle related Disease Predictor," *2022 5th International Conference on Advances in Science and Technology (ICAST).*

[2] Xia Yu and Shuoyu Wang, "A Health Check and Prediction System for Lifestyle-Related Disease Prevention," *First International Conference on Innovative Computing, Information and Control - Volume I (ICICIC'06)*

[3] M. Govindaraj, V. Asha, B. Saju, M. Sagar and Rahul, "Machine Learning Algorithms for Disease Prediction Analysis," *2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)*

[4] "Machine Learning for Healthcare: On the Verge of a Major Shift in Healthcare Epidemiology" by Alaa M. Ahmed et al.

[5] "Machine Learning in Medicine: A Practical Introduction" by Thomas H.McCoy Jr. et al.

[6] "Machine Learning for Predictive Modeling of Health Outcomes" by Rajkomar et al.

[7] PubMed Central (PMC) - Machine Learning and Healthcare Section.

[8] Scikit-learn Documentation https://scikit-learn.org/

[9] Flask Documentation https://flask.palletsprojects.com/

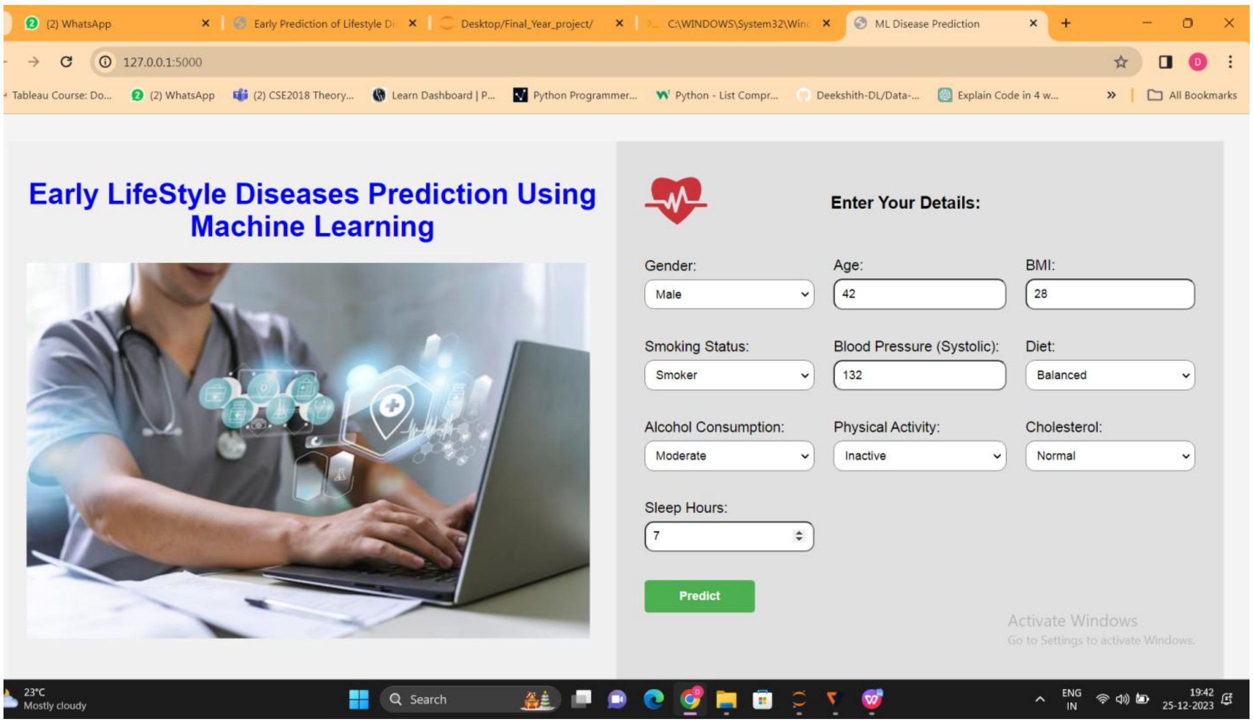
\

Appendix A

Screenshots

Test Cases:-1

A.1 Heart Disease:



Early LifeStyle Diseases Prediction Using Machine Learning

Enter Your Details:

Gender: Male

Age: 42

BMI: 28

Smoking Status: Smoker

Blood Pressure (Systolic): 132

Diet: Balanced

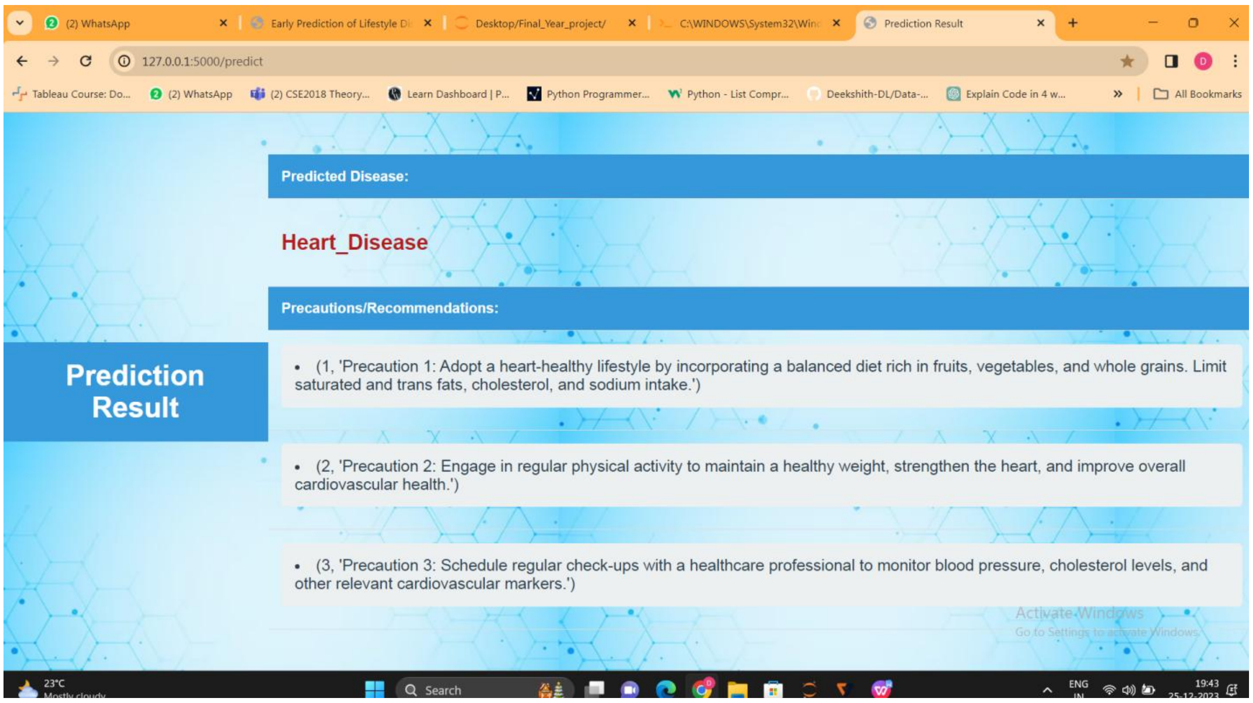
Alcohol Consumption: Moderate

Physical Activity: Inactive

Cholesterol: Normal

Sleep Hours: 7

User Input for Heart_Diseases



Predicted Disease:

Heart_Disease

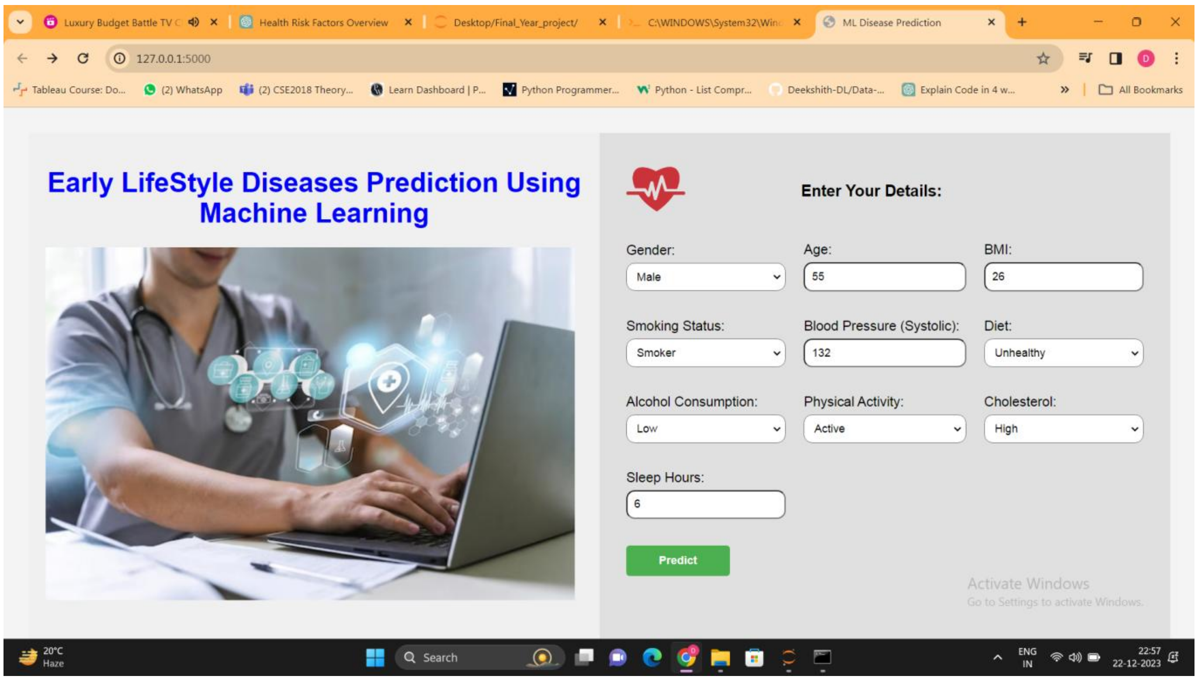
Precautions/Recommendations:

- (1, 'Precaution 1: Adopt a heart-healthy lifestyle by incorporating a balanced diet rich in fruits, vegetables, and whole grains. Limit saturated and trans fats, cholesterol, and sodium intake.')
- (2, 'Precaution 2: Engage in regular physical activity to maintain a healthy weight, strengthen the heart, and improve overall cardiovascular health.')
- (3, 'Precaution 3: Schedule regular check-ups with a healthcare professional to monitor blood pressure, cholesterol levels, and other relevant cardiovascular markers.')

Output for Heart_Diseases

Test Cases:-2

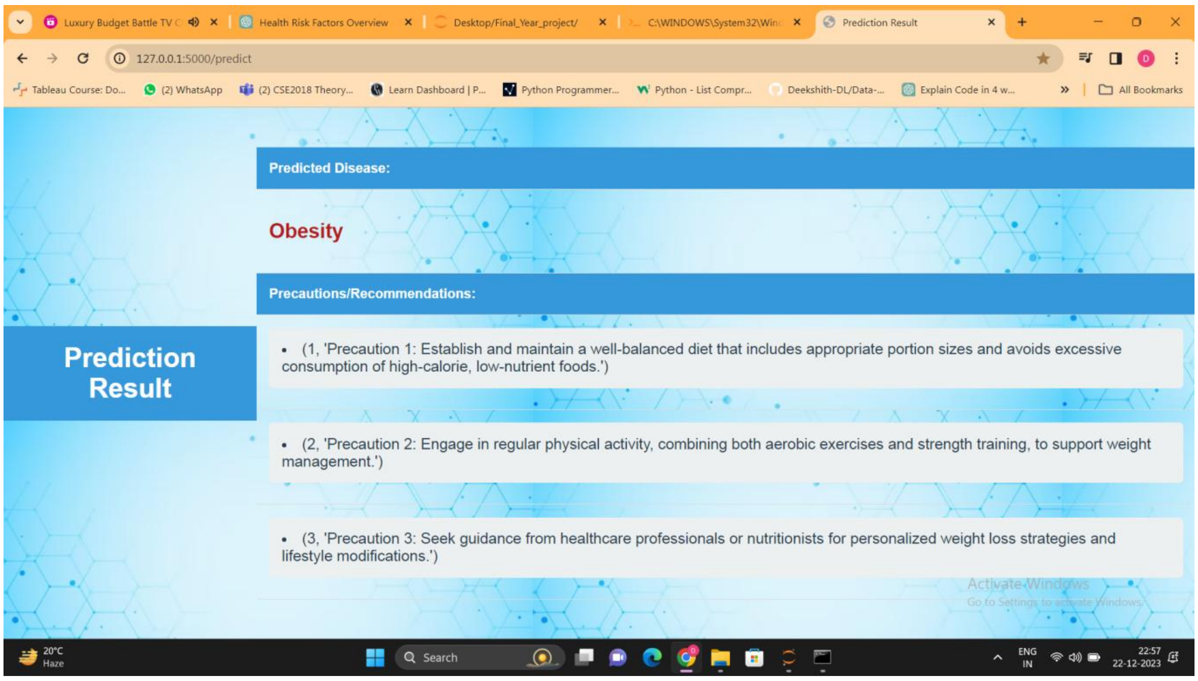
9.2 Obesity:-1


Early LifeStyle Diseases Prediction Using Machine Learning

Enter Your Details:

Gender: Male
Age: 55
BMI: 26
Smoking Status: Smoker
Blood Pressure (Systolic): 132
Diet: Unhealthy
Alcohol Consumption: Low
Physical Activity: Active
Cholesterol: High
Sleep Hours: 6

User Input for Obesity


Prediction Result

Predicted Disease:

Obesity

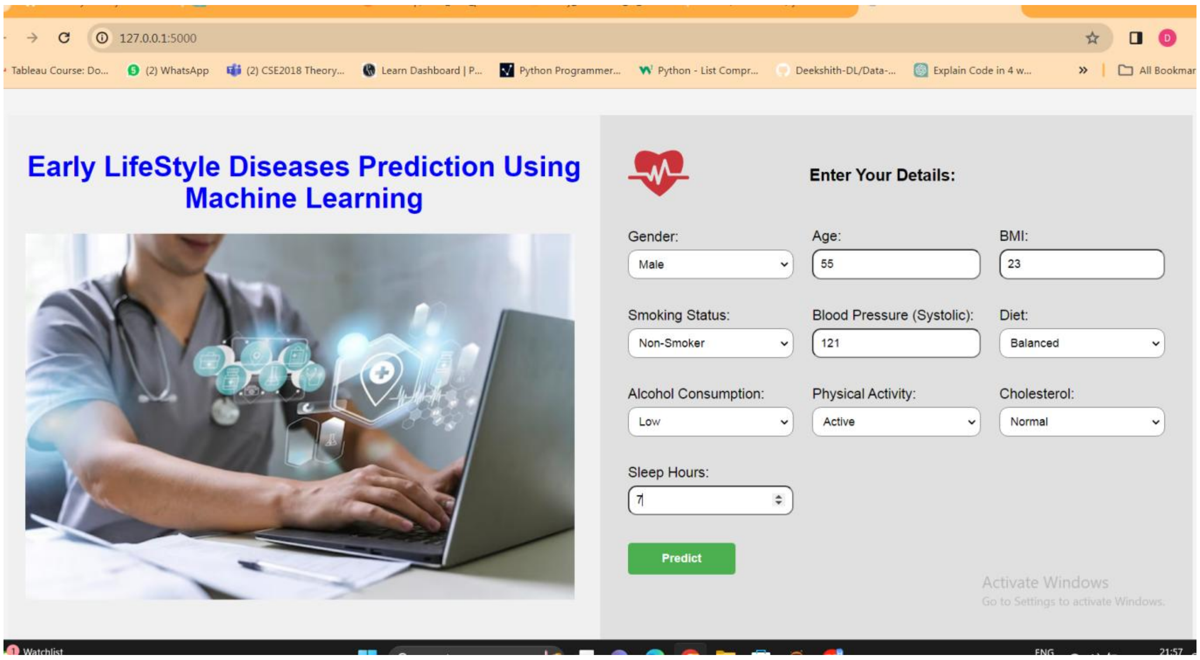
Precautions/Recommendations:

- (1, 'Precaution 1: Establish and maintain a well-balanced diet that includes appropriate portion sizes and avoids excessive consumption of high-calorie, low-nutrient foods.')
- (2, 'Precaution 2: Engage in regular physical activity, combining both aerobic exercises and strength training, to support weight management.')
- (3, 'Precaution 3: Seek guidance from healthcare professionals or nutritionists for personalized weight loss strategies and lifestyle modifications.')

Output for Obesity

Test Cases:-3

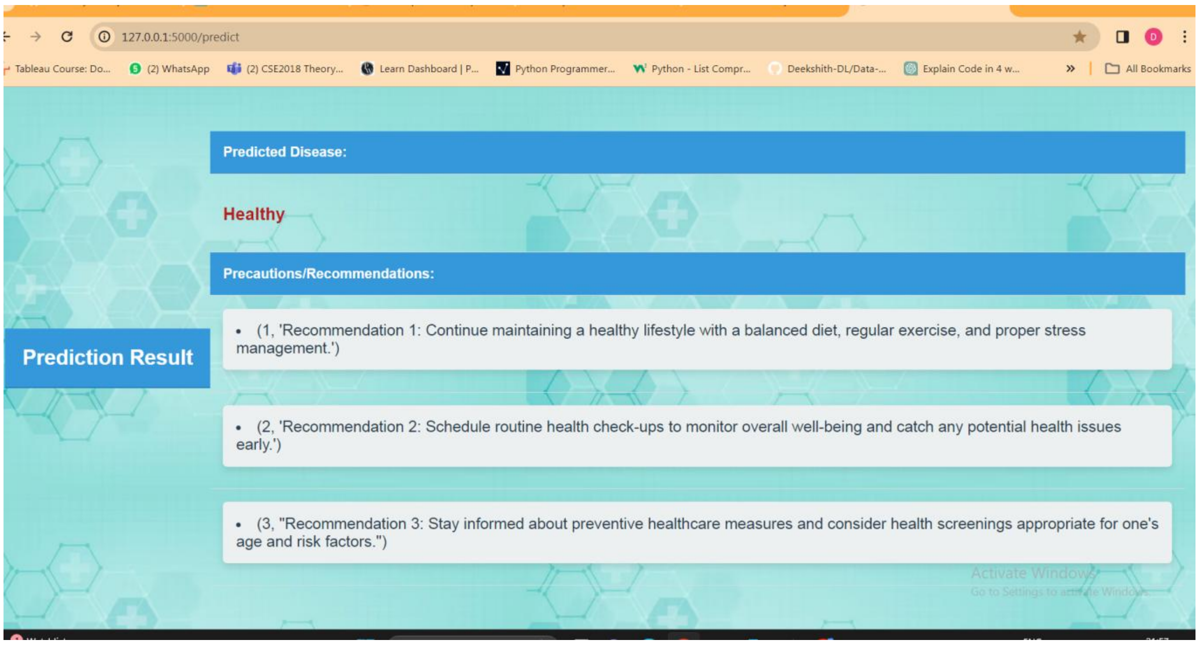
9.3 Healthy:


Early LifeStyle Diseases Prediction Using Machine Learning

Enter Your Details:

Gender: Male
Age: 55
BMI: 23
Smoking Status: Non-Smoker
Blood Pressure (Systolic): 121
Diet: Balanced
Alcohol Consumption: Low
Physical Activity: Active
Cholesterol: Normal
Sleep Hours: 7

User Input for Healthy


Prediction Result

Predicted Disease:

Healthy

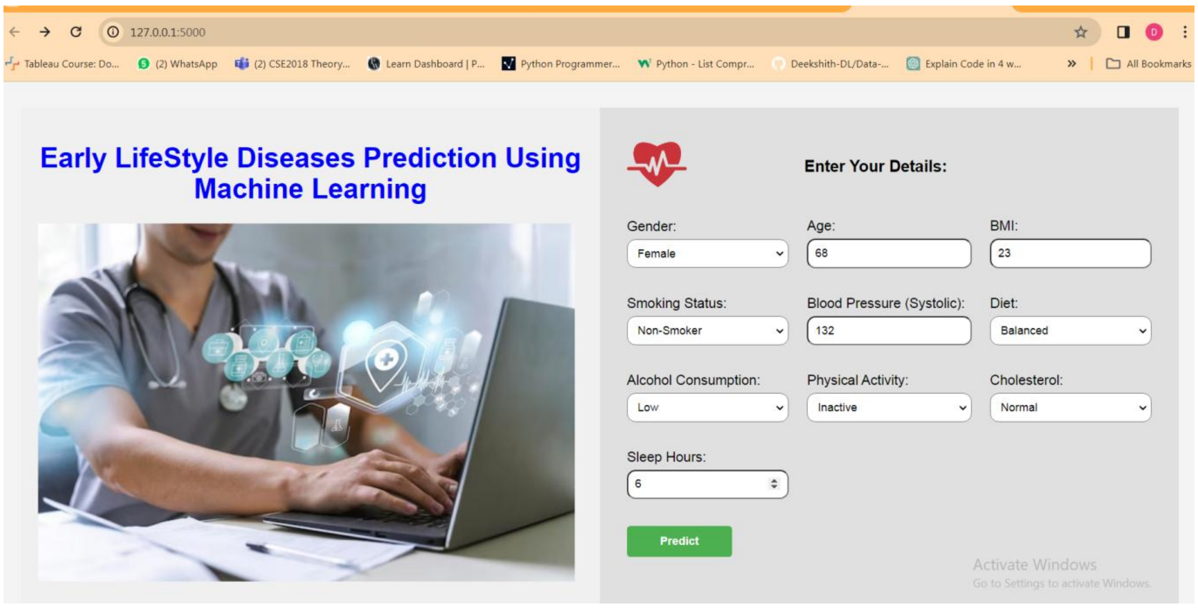
Precautions/Recommendations:

- (1, 'Recommendation 1: Continue maintaining a healthy lifestyle with a balanced diet, regular exercise, and proper stress management.')
- (2, 'Recommendation 2: Schedule routine health check-ups to monitor overall well-being and catch any potential health issues early.')
- (3, "Recommendation 3: Stay informed about preventive healthcare measures and consider health screenings appropriate for one's age and risk factors.")

Output for Healthy

Test Cases:-4

9.4 Hypertension:

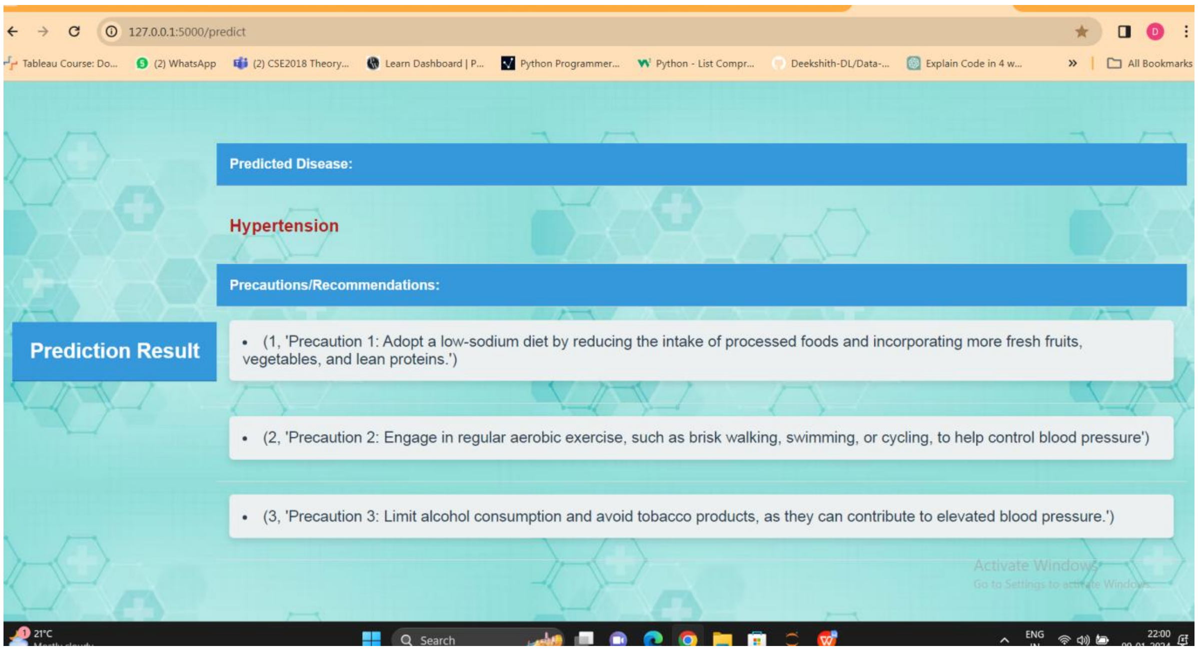


Early LifeStyle Diseases Prediction Using Machine Learning

Enter Your Details:

Gender: Female
Age: 68
BMI: 23
Smoking Status: Non-Smoker
Blood Pressure (Systolic): 132
Diet: Balanced
Alcohol Consumption: Low
Physical Activity: Inactive
Cholesterol: Normal
Sleep Hours: 6

User Input for Hypertension



Predicted Disease:

Hypertension

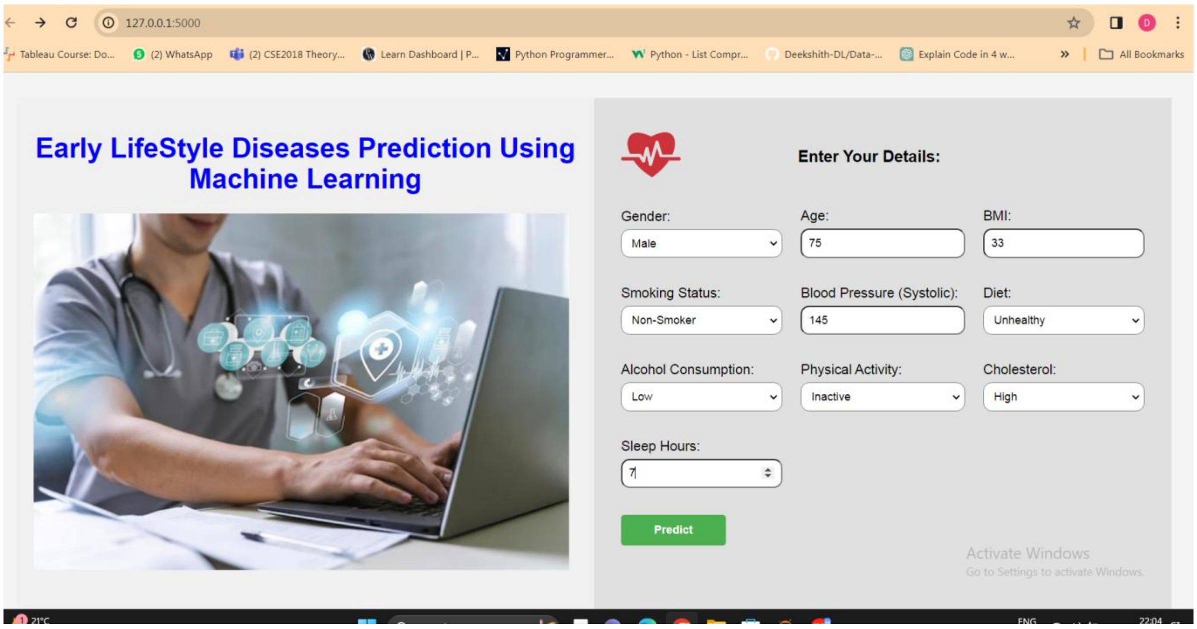
Precautions/Recommendations:

- (1, 'Precaution 1: Adopt a low-sodium diet by reducing the intake of processed foods and incorporating more fresh fruits, vegetables, and lean proteins.')
- (2, 'Precaution 2: Engage in regular aerobic exercise, such as brisk walking, swimming, or cycling, to help control blood pressure')
- (3, 'Precaution 3: Limit alcohol consumption and avoid tobacco products, as they can contribute to elevated blood pressure.')

Output for Hypertension

Test Cases:-5

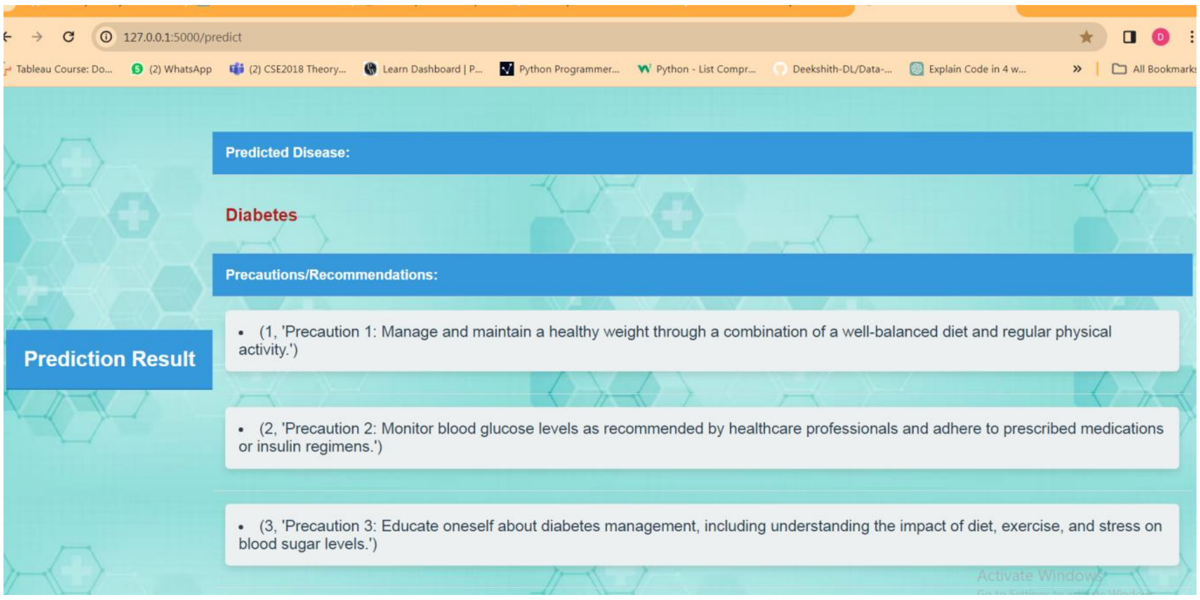
9.5 Diabetes:


Early LifeStyle Diseases Prediction Using Machine Learning

Enter Your Details:

Gender: Male
Age: 75
BMI: 33
Smoking Status: Non-Smoker
Blood Pressure (Systolic): 145
Diet: Unhealthy
Alcohol Consumption: Low
Physical Activity: Inactive
Cholesterol: High
Sleep Hours: 7

User Input for Diabetes


Predicted Disease:

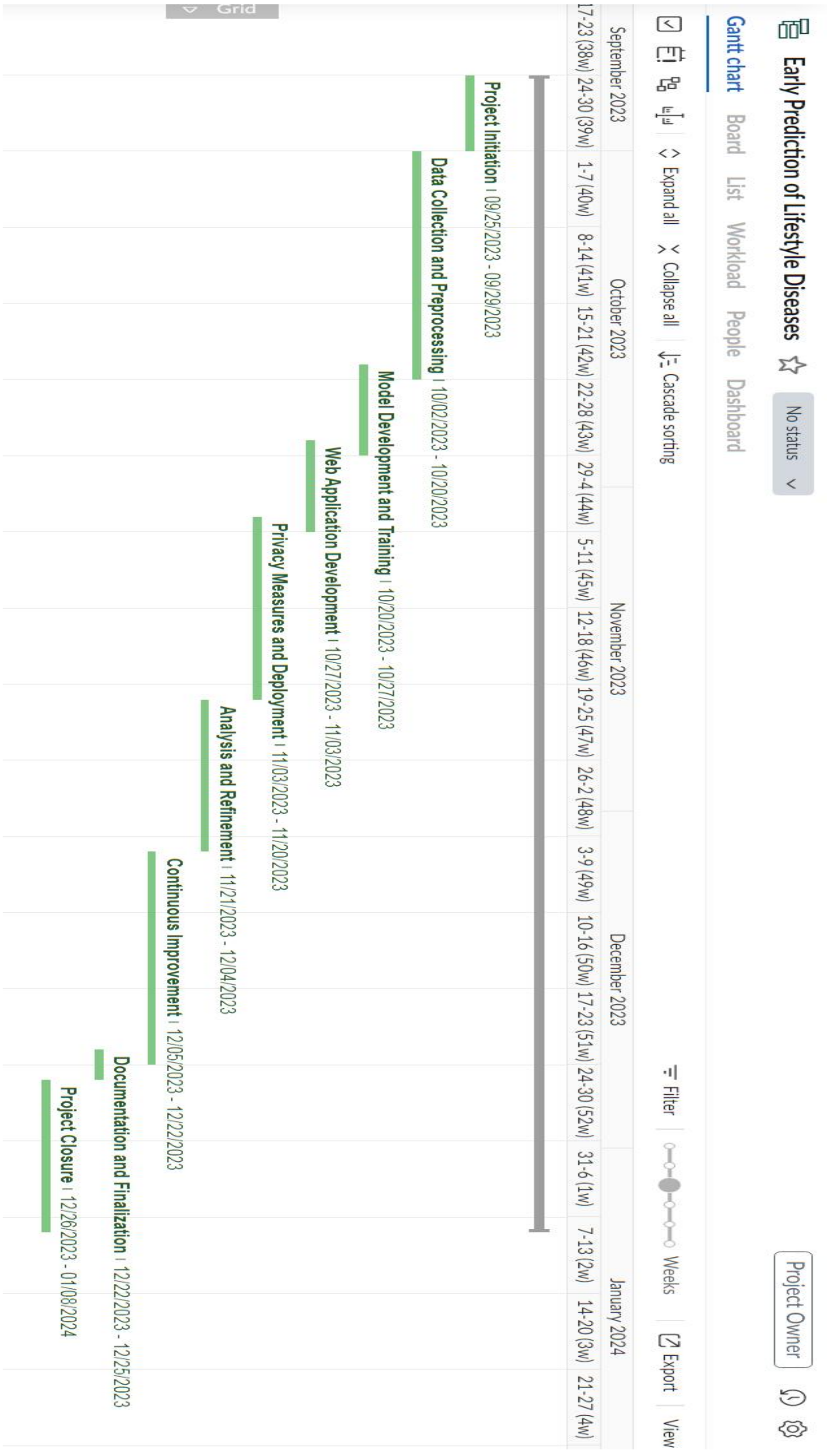
Diabetes

Precautions/Recommendations:

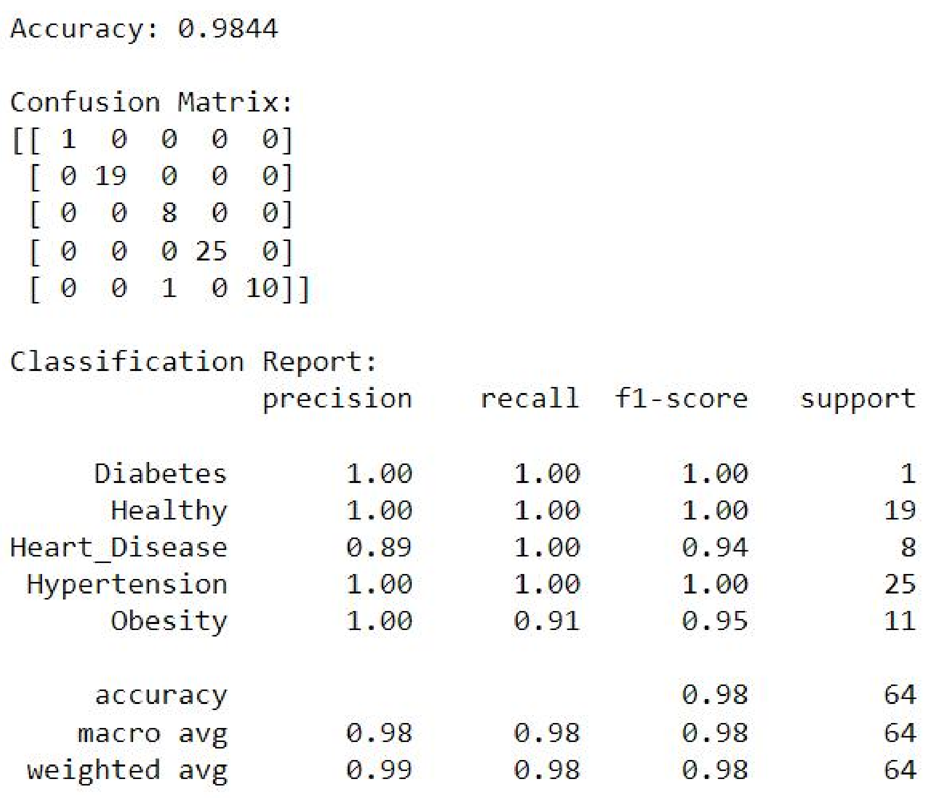
- (1, 'Precaution 1: Manage and maintain a healthy weight through a combination of a well-balanced diet and regular physical activity.')
- (2, 'Precaution 2: Monitor blood glucose levels as recommended by healthcare professionals and adhere to prescribed medications or insulin regimens.')
- (3, 'Precaution 3: Educate oneself about diabetes management, including understanding the impact of diet, exercise, and stress on blood sugar levels.')

Output for Diabetes

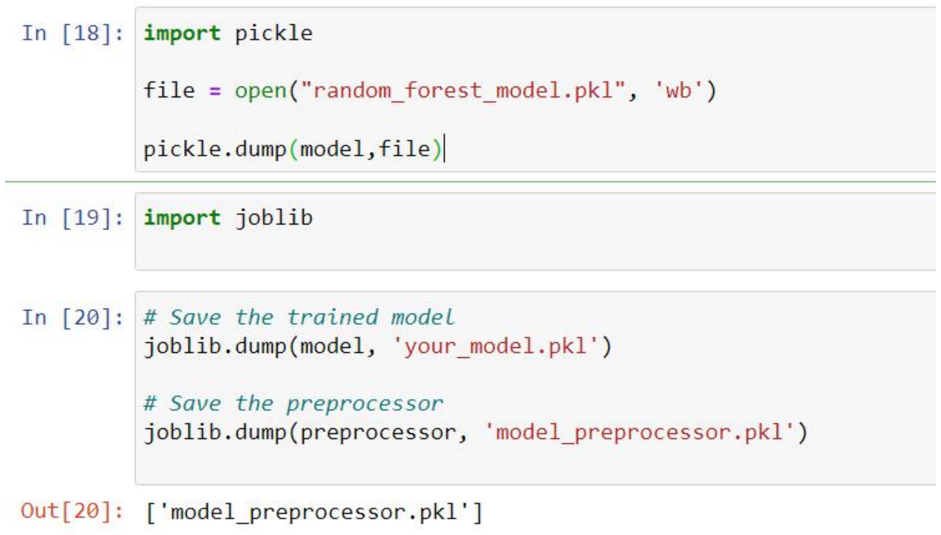
Detailed Gantt Chart



Metrices:



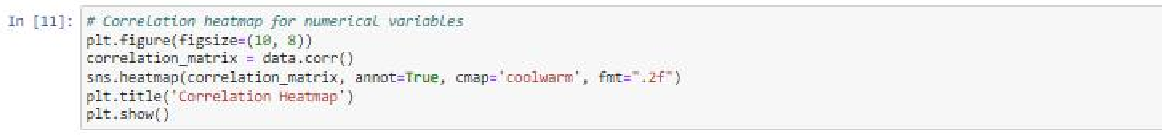
```
Accuracy: 0.9844

Confusion Matrix:
[[ 1  0  0  0  0]
 [ 0 19  0  0  0]
 [ 0  0  8  0  0]
 [ 0  0  0 25  0]
 [ 0  0  1  0 10]]

Classification Report:
               precision    recall  f1-score   support

     Diabetes       1.00      1.00      1.00         1
      Healthy       1.00      1.00      1.00        19
Heart_Disease       0.89      1.00      0.94         8
 Hypertension       1.00      1.00      1.00        25
      Obesity       1.00      0.91      0.95        11

     accuracy                           0.98        64
    macro avg       0.98      0.98      0.98        64
 weighted avg       0.99      0.98      0.98        64
```



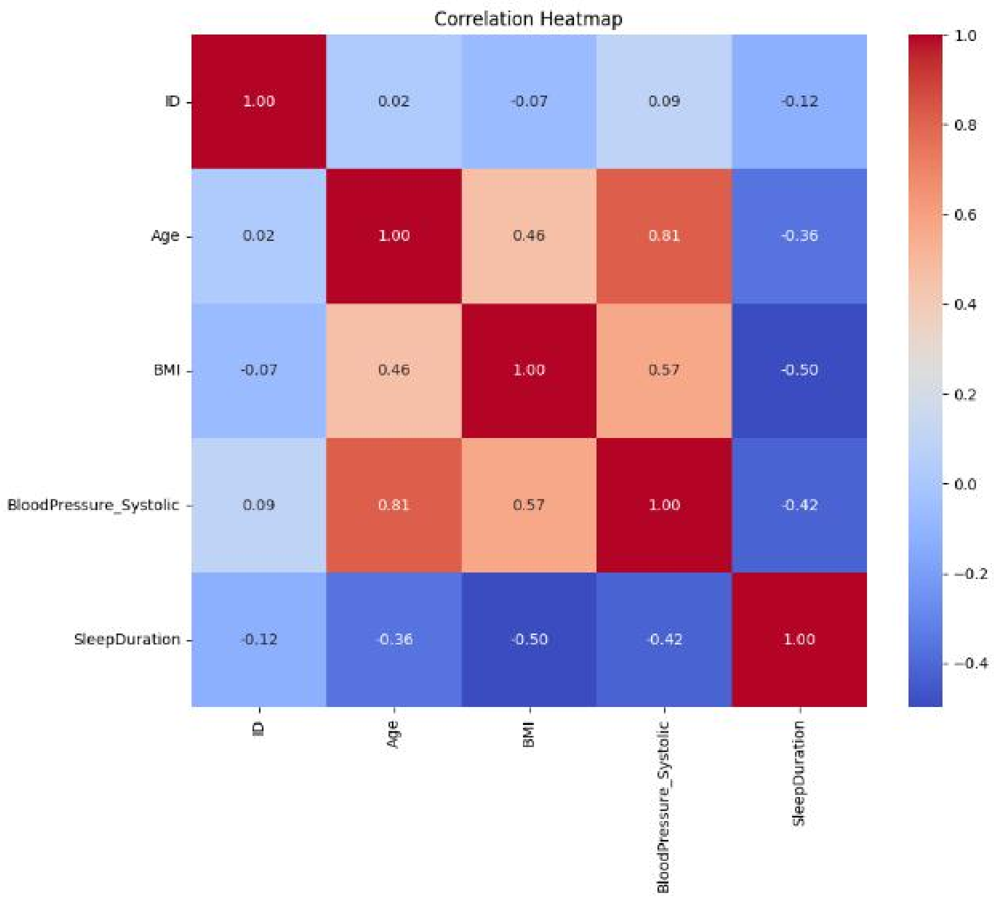
```
In [18]: import pickle

         file = open("random_forest_model.pkl", 'wb')

         pickle.dump(model,file)
```

```
In [19]: import joblib
```

```
In [20]: # Save the trained model
         joblib.dump(model, 'your_model.pkl')

         # Save the preprocessor
         joblib.dump(preprocessor, 'model_preprocessor.pkl')

Out[20]: ['model_preprocessor.pkl']
```



```
In [11]: # Correlation heatmap for numerical variables
         plt.figure(figsize=(10, 8))
         correlation_matrix = data.corr()
         sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
         plt.title('Correlation Heatmap')
         plt.show()
```



Heat Map for Numerical Features

Project GitHub Link:

Link: https://github.com/Deekshith-DL/ML-Early-LifeStyle-Diseases-Prediction-Web-App

Appendix B
Pseudocode and Algorithm

A. App.py

[i]. Pseudocode

```
1. Import necessary libraries and modules (Flask, pandas, sklearn, joblib).
2. Load the machine learning model and preprocessor using joblib.
3. Define a function `get_precautions` to provide precautions based on the predicted disease.
4. Define a function `get_chatbot_response` to implement chatbot logic (for future extension).
5. Define the Flask application (`app`) and routes:
   a. Home route: Render the home page.
   b. Chat route: Process user messages and return chatbot responses.
   c. Predict route: Handle user input, preprocess it, predict disease, and render result page.
6. Run the Flask application if the script is executed directly.
```

[ii]. Algorithm

```
1. Import necessary libraries and modules:
   - Flask for web application
   - pandas for data manipulation
   - sklearn for machine learning
   - joblib for model and preprocessor loading

2. Load the machine learning model and preprocessor using joblib:
   - Load the trained model (e.g., "your_model.pkl").
   - Load the preprocessor (e.g., "model_preprocessor.pkl").

3. Define a function `get_precautions(predicted_disease)`:
   - Provide precautions based on the predicted disease.
   - Return a list of precautions.
```

```
4. Define a function `get_chatbot_response(user_message)`:
  - Implement chatbot logic (for future extension).
  - For now, echo the user's message.

5. Define the Flask application (`app`) and routes:
  a. Home route (`'/'`):
    - Render the home page using the 'index.html' template.

  b. Chat route (`'/chat'`):
    - Receive JSON data containing the user's message.
    - Process the message using `get_chatbot_response`.
    - Return the chatbot response in JSON format.

  c. Predict route (`'/predict'`):
    - Receive user input from a form.
    - Create a DataFrame from the input.
    - Preprocess the input using the preprocessor.
    - Predict the disease using the loaded model.
    - Get precautions based on the predicted disease using `get_precautions`.
    - Render the result page ('result.html') with the prediction and precautions.

6. Run the Flask application if the script is executed directly:
  - Start the Flask application with debugging enabled.
```

B. Index.html

[i]. Pseudocode:

```
1. Set up HTML document structure with necessary metadata and stylesheet link.
2. Create a container with two sides (left and right).
3. Left-side content:
  a. Display a title and an image for the application.
4. Right-side content:
  a. Display an image for user details entry.
  b. Create a form for user input with the following fields:
    - Gender (dropdown)
    - Age (number input)
    - BMI (number input)
    - Smoking Status (dropdown)
    - Blood Pressure (Systolic) (number input)
    - Diet (dropdown)
    - Alcohol Consumption (dropdown)
    - Physical Activity (dropdown)
    - Cholesterol (dropdown)
    - Sleep Hours (number input)
    - Predict button (submit form)
  c. Display the prediction result if available.
5. Apply styling to the HTML elements for a visually appealing layout.
```

[ii]. Algorithm:

```
1. Set up HTML document structure:
  a. Declare document type, language, character set, and viewport.
  b. Link external stylesheet (style.css).
  c. Set the title of the document.

2. Create a container with two sides:
  a. Left-side content:
    - Display a heading and an image related to the application.
  b. Right-side content:
    - Display an image related to user details entry.
    - Create a form with appropriate fields for user input:
      - Gender (dropdown)
      - Age (number input)
      - BMI (number input)
      - Smoking Status (dropdown)
      - Blood Pressure (Systolic) (number input)
      - Diet (dropdown)
      - Alcohol Consumption (dropdown)
      - Physical Activity (dropdown)
```

```
        - Cholesterol (dropdown)
        - Sleep Hours (number input)
        - Predict button (submit form)
      - Display the prediction result if available.

3. Apply styling:
   a. Set the background color, font family, and other styles for the body.
   b. Style the container, headings, labels, inputs, selects, buttons, and other elements for a cohesive layout.
   c. Add styles for images and adjust spacing as needed.

4. Save the HTML code as an 'index.html' file.
```

C. Result.html

[i]. Pseudocode:

```
1. Set up HTML document structure with necessary metadata.
2. Create a container with styles for the body.
3. Define styles for the table, headers, headings, and list items.
4. Display the predicted disease and corresponding precautions/recommendations.
```

[ii]. Algorithm:

```
1. Set up HTML document structure:
   a. Declare document type, language, character set, and viewport.
   b. Set the title of the document.

2. Create a container with styles for the body:
   a. Set the font family, background image, text color, margin, padding, display, and alignment.
   b. Use flexbox to center content vertically and horizontally.
   c. Apply additional styles for text alignment.

3. Define styles for the table, headers, headings, and list items:
   a. Set the width, border-collapse, margin, padding, text-align, and border properties for the
```

table.

b. Style headers (th) with background color, text color, padding, font size, and border properties.

c. Style cells (td) with padding, text-align, and border properties.

d. Style the main heading (h1) with background color, text color, padding, margin, font size, and border properties.

e. Style secondary headings (h2) with color, margin-top, and font size.

f. Style unordered list (ul) with list-style-type, padding, and margin.

g. Style list items (li) with font size, margin, color, background color, padding, border radius, and box shadow.

4. Display the predicted disease and corresponding precautions/recommendations:

a. Display the main heading with "Prediction Result."

b. Display a table with two columns for "Predicted Disease" and "Precautions/Recommendations."

c. Use template variables ({{ prediction }} and {% for precaution in precautions %}) to dynamically display the predicted disease and precautions.

d. Use a loop to iterate through the list of precautions and display each as a list item.

5. Save the HTML code as a 'result.html' file.

Domain Knowledge:

Heart Disease:

Risk factors

1.Age. Growing older increases the risk of damaged and narrowed arteries and a weakened or thickened heart muscle.

2.Sex. Men are generally at greater risk of heart disease. ...

3.Family history. ...

4.Smoking. ...

5.Unhealthy diet. ...

6.High blood pressure. ...

7.High cholesterol. ...

8.Diabetes.

Diabetes

1.Have prediabetes.

2.Are overweight.

3.Are 45 years or older.

4.Have a parent, brother, or sister with type 2 diabetes.

5.Are physically active less than 3 times a week.

6.Have ever had gestational diabetes (diabetes during pregnancy) or given birth to a baby who weighed over 9 pounds.

Hypertension

1.older age.

2.genetics.

3.being overweight or obese.

4.not being physically active.

5.high-salt diet.

6.drinking too much alcohol.

Obesity

1.Food and Activity. People gain weight when they eat more calories than they burn through activity. ...

2.Environment. ...

3.Genetics. ...

4.Health Conditions and Medications. ...

5.Stress, Emotional Factors, and Poor Sleep.

Summary

The project aims to develop a predictive healthcare system that utilizes advanced machine learning models to early predict the likelihood of lifestyle diseases in individuals during their routine physical checkups. By leveraging detailed demographic and vital statistics data, this system will enable proactive and preventive healthcare measures, reducing the overall cost of treatment and improving the quality of life.

Project Objectives:

1. Build a comprehensive database of demographic and vital statistics data from a diverse population, including individuals with lifestyle diseases and those without.
2. Develop and train machine learning models capable of predicting specific lifestyle diseases, such as diabetes, heart disease, and hypertension, based on the collected data.
3. Implement a user-friendly and secure web or mobile application to provide easy access for both healthcare providers and individuals to input data and receive disease predictions.
4. Ensure the system's accuracy, reliability, and scalability to accommodate a large number of users and healthcare providers.
5. Collaborate with healthcare professionals and institutions to validate the system's predictions and incorporate feedback for continuous improvement.

Key Components:

1. Data Collection and Preprocessing:

- Gather detailed demographic information (age, gender, occupation, location, etc.) and vital statistics (blood pressure, cholesterol levels, BMI, etc.) from a diverse set of individuals.
- Ensure data privacy and security by anonymizing and encrypting sensitive information.

2. Machine Learning Model Development:

- Create machine learning models, such as logistic regression, decision trees, random forests, or deep neural networks, to predict the likelihood of specific lifestyle diseases.
- Train the models using the collected data and employ techniques for feature selection and engineering to enhance prediction accuracy.

3. User Interface (UI):
- Develop a user-friendly web or mobile application for individuals to input their data during physical checkups.
- Create a separate interface for healthcare providers to access and review patient predictions.

4. Prediction and Recommendations:
- Implement an algorithm that processes user data and provides instant predictions regarding the likelihood of lifestyle diseases.
- Offer personalized recommendations for preventive healthcare strategies, such as diet, exercise, and regular checkups, based on the predictions.

5. Integration with Healthcare Systems:
- Collaborate with healthcare institutions to integrate the predictive system into their existing healthcare infrastructure.
- Ensure data interoperability and compatibility with electronic health records (EHR) systems.

6. Validation and Continuous Improvement:
- Work closely with healthcare professionals to validate the accuracy and reliability of the predictions.
- Continuously update and refine the machine learning models based on new data and feedback from healthcare providers.

Technology Stack:
- Machine Learning: Python, scikit-learn.
- Web Application: HTML, CSS.
- Deployment: Flask
- Environment/platform: Jupyter-notebook
- Host: Web Browser

Future Work

1. Integration with Real-Time Data:

- Enhance the system to integrate real-time health data from wearable devices or health monitoring systems. This would provide more accurate and up-to-date predictions.

2. Incorporation of More Health Parameters:

- Expand the model to consider a broader range of health parameters. This might include genetic information, additional blood markers, or lifestyle factors for a more comprehensive prediction.

3. Enhanced User Interface (UI) and User Experience (UX):

- Improve the user interface to make it more intuitive and user-friendly. Consider incorporating data visualization techniques to help users better understand the predictions and recommendations.

4. Multi-Modal Data Processing:

- Extend the model to handle multi-modal data, such as combining textual health descriptions, images, or voice data. This can provide a more holistic view of an individual's health status.

5. Geographical and Demographic Customization:

- Tailor the prediction model based on geographical and demographic factors. Health risks can vary across different regions and population groups, and customizing the model could enhance its accuracy.

6. Integration of Telemedicine and Healthcare Providers:

- Collaborate with healthcare providers to integrate the prediction system into telemedicine platforms. This would enable seamless communication between users and healthcare professionals for better health management.

7. Longitudinal Health Tracking:

- Implement a system for longitudinal health tracking, allowing users to monitor changes in their health over time. This could be beneficial for early detection of trends or patterns that might indicate potential health issues.

8. User Feedback and Learning:

- Implement a feedback loop where users can provide feedback on the predictions and recommendations. This data can be used to continuously improve and refine the prediction model through machine learning techniques.

9. Enhanced Privacy Measures:

- Strengthen privacy measures to ensure the secure handling of user health data. This is especially important as the system deals with sensitive health information.

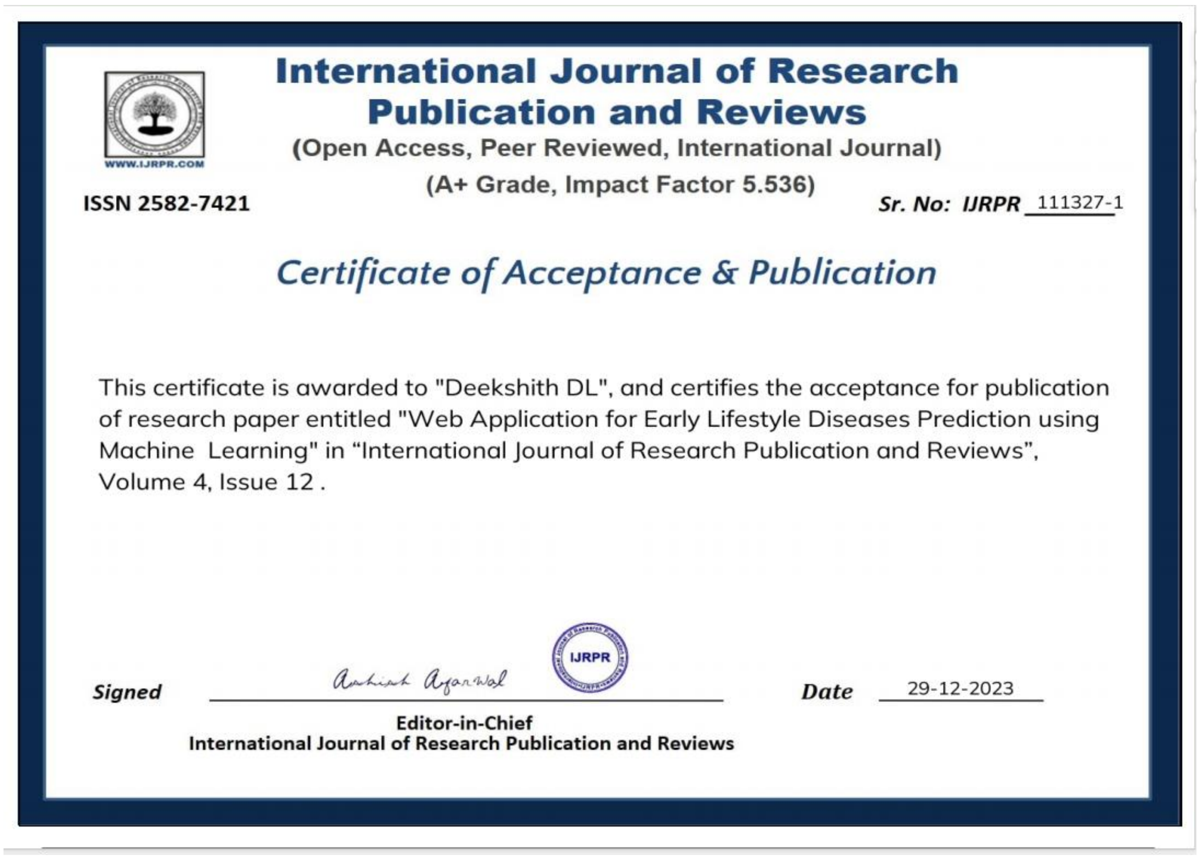
10. Expand Disease Categories:

- Include more lifestyle-related diseases and conditions in the prediction model. This expansion would provide users with a broader understanding of their health risks.

11. Community Engagement and Education:

- Develop features for community engagement and health education. This could include forums, blogs, or educational content to empower users to take control of their health.

Paper Publication Certification



WWW.IJRPR.COM

International Journal of Research Publication and Reviews

(Open Access, Peer Reviewed, International Journal)

(A+ Grade, Impact Factor 5.536)

ISSN 2582-7421

Sr. No: IJRPR 111327-1

Certificate of Acceptance & Publication

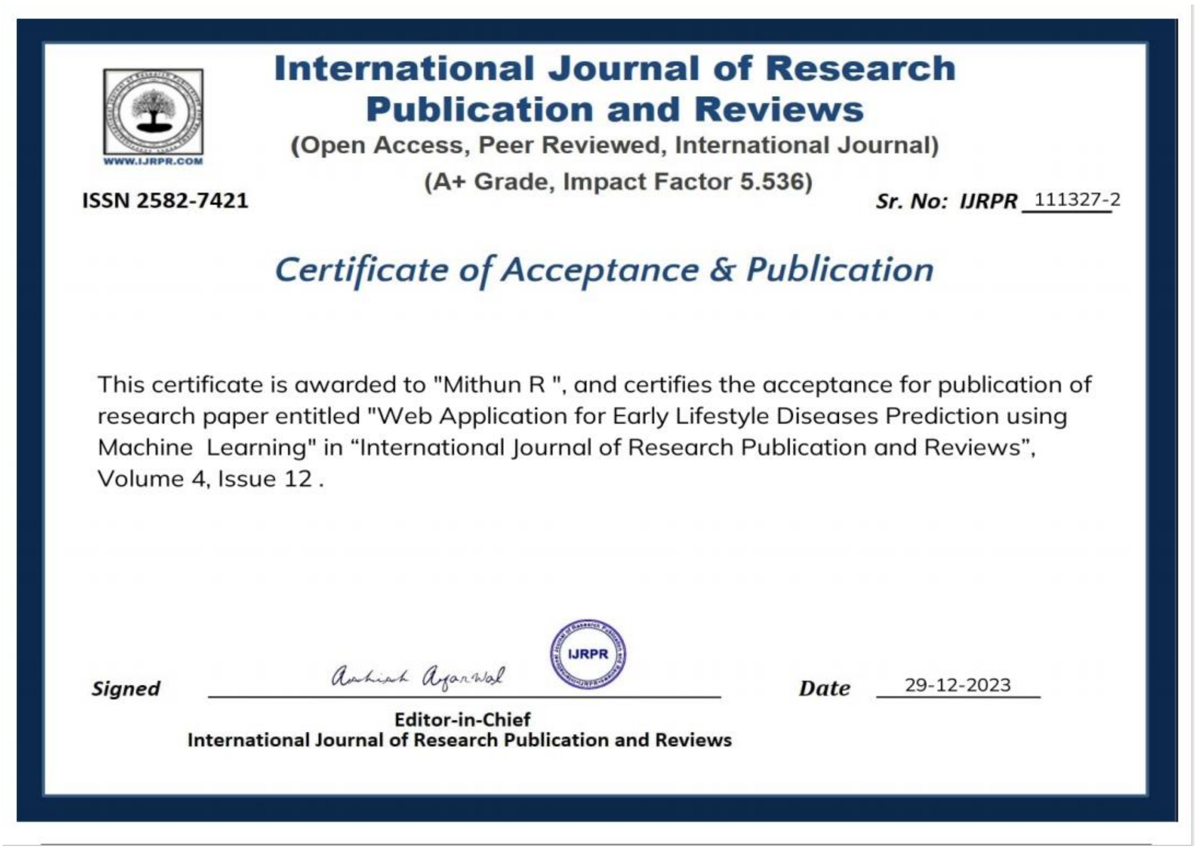
This certificate is awarded to "Deekshith DL", and certifies the acceptance for publication of research paper entitled "Web Application for Early Lifestyle Diseases Prediction using Machine Learning" in "International Journal of Research Publication and Reviews", Volume 4, Issue 12 .

Signed [illegible] IJRPR

Editor-in-Chief
International Journal of Research Publication and Reviews

Date 29-12-2023

Paper Publication Certification



International Journal of Research Publication and Reviews

(Open Access, Peer Reviewed, International Journal)

(A+ Grade, Impact Factor 5.536)

WWW.IJRPR.COM

ISSN 2582-7421

Sr. No: IJRPR 111327-2

Certificate of Acceptance & Publication

This certificate is awarded to "Mithun R ", and certifies the acceptance for publication of research paper entitled "Web Application for Early Lifestyle Diseases Prediction using Machine Learning" in “International Journal of Research Publication and Reviews”, Volume 4, Issue 12 .

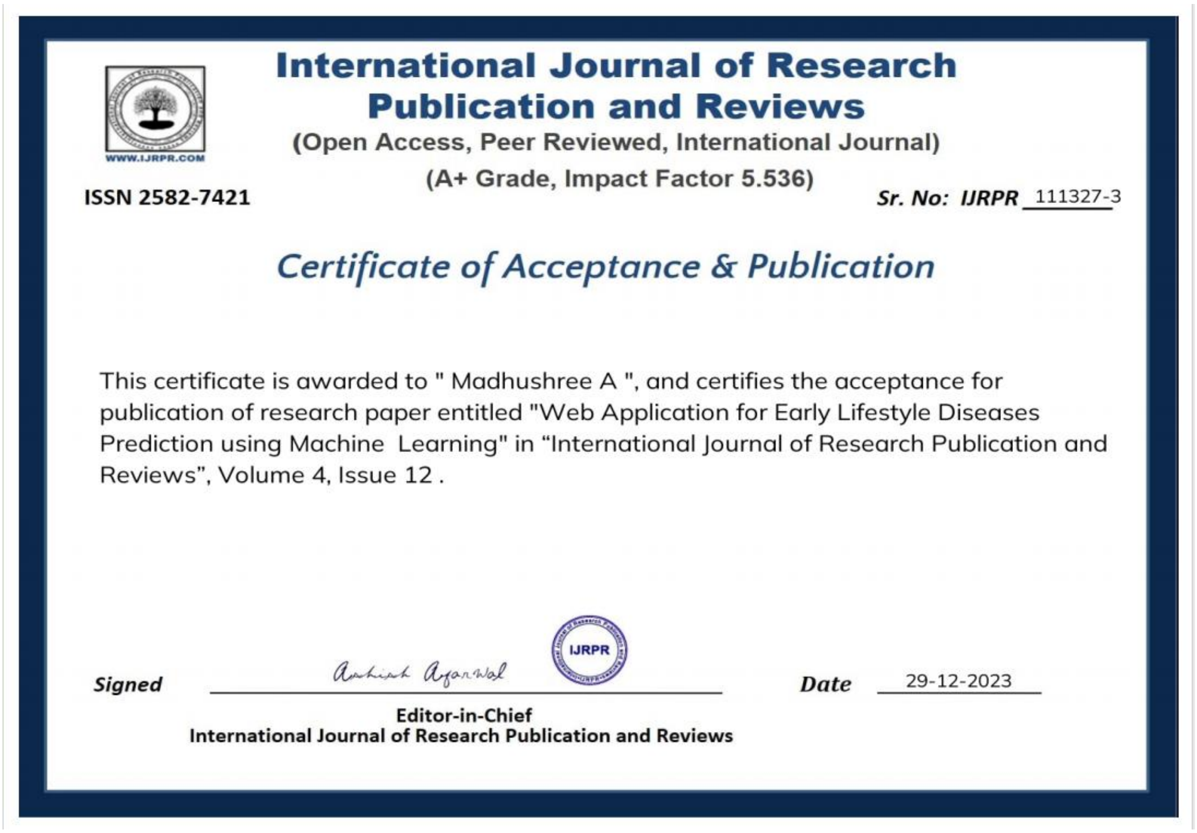
Signed Aashish Agarwal

IJRPR

Editor-in-Chief
International Journal of Research Publication and Reviews

Date 29-12-2023

Paper Publication Certification



International Journal of Research Publication and Reviews

(Open Access, Peer Reviewed, International Journal)

(A+ Grade, Impact Factor 5.536)

WWW.IJRPR.COM

ISSN 2582-7421

Sr. No: IJRPR 111327-3

Certificate of Acceptance & Publication

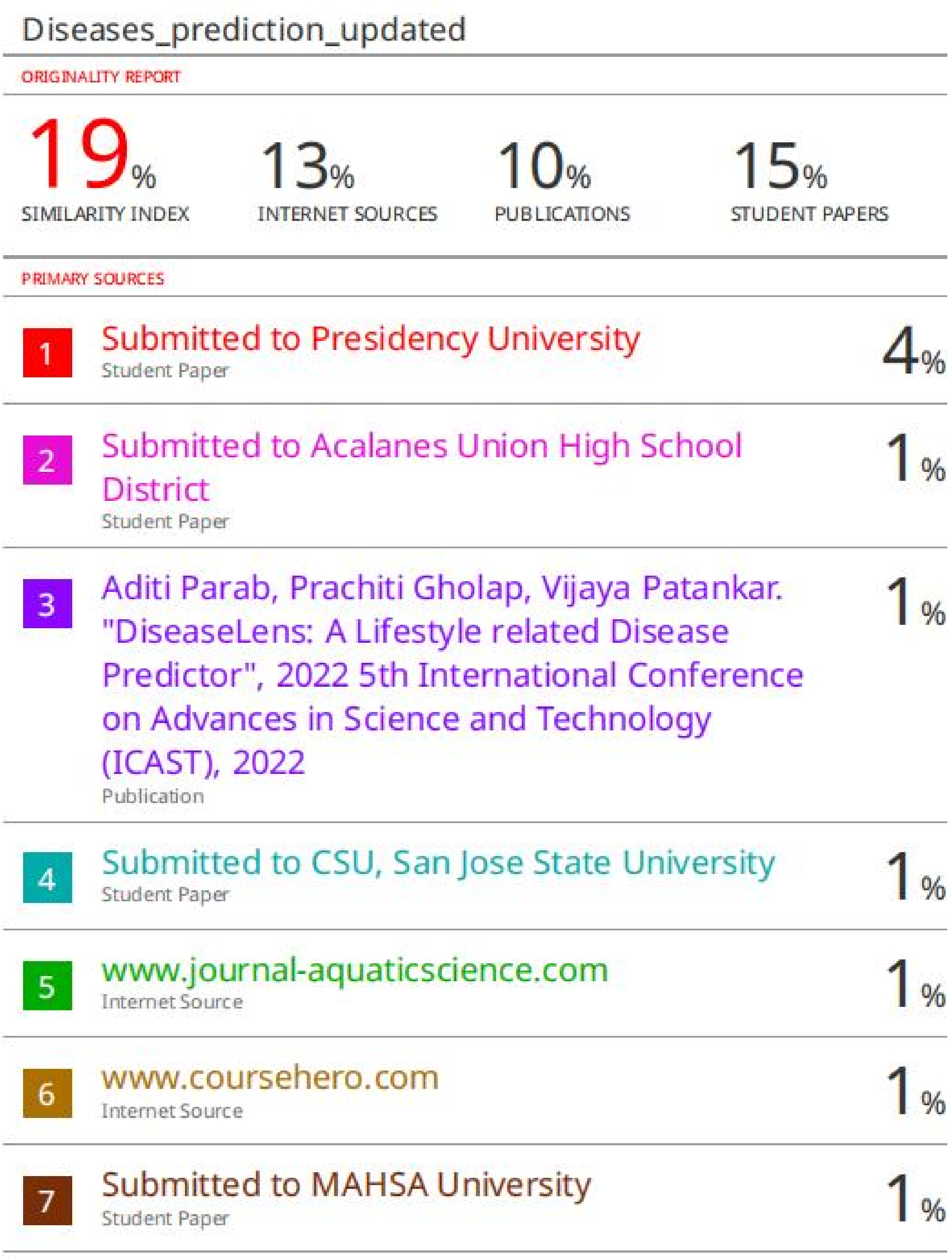
This certificate is awarded to " Madhushree A ", and certifies the acceptance for publication of research paper entitled "Web Application for Early Lifestyle Diseases Prediction using Machine Learning" in "International Journal of Research Publication and Reviews", Volume 4, Issue 12 .

Signed Ashish Agarwal IJRPR

Date 29-12-2023

Editor-in-Chief
International Journal of Research Publication and Reviews

Plagiarism Report



Diseases_prediction_updated

ORIGINALITY REPORT

19%	13%	10%	15%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Presidency University Student Paper	4%
2	Submitted to Acalanes Union High School District Student Paper	1%
3	Aditi Parab, Prachiti Gholap, Vijaya Patankar. "DiseaseLens: A Lifestyle related Disease Predictor", 2022 5th International Conference on Advances in Science and Technology (ICAST), 2022 Publication	1%
4	Submitted to CSU, San Jose State University Student Paper	1%
5	www.journal-aquaticscience.com Internet Source	1%
6	www.coursehero.com Internet Source	1%
7	Submitted to MAHSA University Student Paper	1%

Sustainable Development Goals

SUSTAINABLE DEVELOPMENT GOALS
17 GOALS TO TRANSFORM OUR WORLD
1 NO POVERTY
2 ZERO HUNGER
3 GOOD HEALTH AND WELL-BEING
4 QUALITY EDUCATION
5 GENDER EQUALITY
6 CLEAN WATER AND SANITATION
7 AFFORDABLE AND CLEAN ENERGY
8 DECENT WORK AND ECONOMIC GROWTH
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE
10 REDUCED INEQUALITIES
11 SUSTAINABLE CITIES AND COMMUNITIES
12 RESPONSIBLE CONSUMPTION AND PRODUCTION
13 CLIMATE ACTION
14 LIFE BELOW WATER
15 LIFE ON LAND
16 PEACE, JUSTICE AND STRONG INSTITUTIONS
17 PARTNERSHIPS FOR THE GOALS
SUSTAINABLE DEVELOPMENT GOALS

The project work carried out here is mapping to SGD-3
"GOOD HEALTH AND WELL-BEING".

The project work carried here contributes to the well-being of the human Society. This can be used to analyzing the Early Life Style Diseases Prediction in the early stage so, that the user can take precautions or recommendation . Hence, they can avoid the further consequences which might be advance for the Well-Being.