



MyMedCare : Disease Prediction using ML & DL

Submitted In Partial Fulfillment of Requirements
For the Degree Of

**Bachelor of Science
(Data Science)**

By

Tanvi Vinod Deodhar

Roll No: 31013021001

Guide

Ms. Minal Dive



S K Somaiya College
Somaiya Vidyavihar University
Vidyavihar, Mumabi - 400 077
2021-24

Somaiya Vidyavihar University

S K Somaiya College

Certificate

This is to certify that the project report on dissertation entitled MyMedCare : Disease Prediction using ML & DL is bonafide record of the dissertation work done by Ms. Tanvi V. Deodhar in the year 2021-2024 under the guidance of Ms. Minal Dive Department of Computer Science in partial fulfillment of requirement for the Bachelor of Science degree in Data Science of Somaiya Vidyavihar University.

Guide/Co-guide

Co-ordinator of the Deparment

Head of the Department

Director

Date:

Place: Mumbai-77

Somaiya Vidyavihar University

S K Somaiya College

Certificate of Approval of Examiners

This is to certify that the project report on dissertation entitled MyMedCare : Disease Prediction using ML & DL is bonafide record of the dissertation work done by Ms. Tanvi V. Deodhar in the year 2021-2024 under the guidance of Ms. Minal Dive Department of Computer Science in partial fulfillment of requirement for the Bachelor of Science degree in Data Science of Somaiya Vidyavihar University.

External Examiner /Expert

Internal Examiner/ Guide

Date:

Place: Mumbai-77

Somaiya Vidyavihar University

S K Somaiya College

DECLARATION

I declare that this written report submission represents the work done based on my and / or others' ideas with adequately cited and referenced the original source. I also declare that I have adhered to all principles of academic honesty and integrity as I have not misinterpreted or fabricated or falsified any idea/data/fact/source/original work/ matter in my submission.

I understand that any violation of the above will be cause for disciplinary action by the college and may evoke the penal action from the sources which have not been properly cited or from whom proper permission is not sought.

Signature of the Student

Tanvi Vinod Deodhar

**31013021001
Roll No.**

Date:

Place: Mumbai-77

Acknowledgment

I would like to express my sincere gratitude to everyone who contributed to the successful completion of my project work.

Their support, guidance, and encouragement have been invaluable throughout this journey.

First and foremost, I extend my heartfelt thanks to my supervisor, MS, Minal Dive and Ms. Swati Maurya for her unwavering support and mentorship. Their expertise, constructive feedback, and dedication played a pivotal role in shaping the direction of my project. I am truly grateful for their time and commitment.

I would like to acknowledge the DEPARTMENT OF COMPUTER SCIENCE and its Head, Ms. Swati Maurya, for providing a conducive academic environment. The resources and facilities made available greatly facilitated my research work. My sincere appreciation goes to all faculty members who shared their knowledge and insights during the course of my study. THANK YOU.

Tanvi Deodhar

Contents

1	Introduction		
	1.1	Introduction	1
	1.2	About the Disease	3
	1.3	Scope and Objective	3
	1.4	Significance of the Study	3
2	Literature Survey		14
	2.1	Tools and Techniques (VS code)	14
	2.2	Machine Learning Models	18
3	Mathematical Background/ Logical Analysis / Fundamental Concepts		31
	3.1	Introduction	31
	3.2	Disease Prediction Model	31
		3.2.1 Heart disease	31
		3.2.2 Breast Cancer	33
		3.2.3 Diabetes Prediction	35
		3.2.4 Parkinsons Disease	37
		3.2.5 Pneumonia	39
4	Coding		42
5	Implementation and Working		111
6	Conclusion		117
7	Future Work		118

Chapter 1

Introduction

In an era marked by the convergence of technology and healthcare, the development of innovative software solutions has emerged as a cornerstone in the quest for efficient diagnosis and treatment. Among these advancements stands MyMedCare, a pioneering software designed to revolutionize the way individuals interact with their health data. MyMedCare is a comprehensive tool that harnesses the power of predictive analysis to offer personalized insights into potential ailments, thereby facilitating early intervention and informed decision-making.

Understanding MyMedCare

MyMedCare operates on a simple yet powerful premise: by inputting symptoms, users receive an output detailing potential diseases matched to their condition. Additionally, the software suggests relevant healthcare professionals, ensuring users can seek appropriate medical attention promptly. Beyond immediate assistance, MyMedCare provides users with a Disease Index, a repository of information covering prevalent diseases such as breast cancer, diabetes, heart disease, Parkinson's, and pneumonia.

Focus Diseases

This thesis centers on the predictive capabilities of MyMedCare, with a particular emphasis on its efficacy in predicting five critical diseases: breast cancer, diabetes, heart disease, Parkinson's, and pneumonia. Each of these conditions poses significant health risks globally, and early detection is paramount for effective management and improved outcomes.

Scope and Objectives

The primary objective of this thesis is to evaluate the accuracy, reliability, and usability of MyMedCare in predicting the aforementioned diseases based on user-provided symptoms. Through rigorous analysis and empirical study, this research aims to:

1. Assess the predictive accuracy of MyMedCare in identifying instances of breast cancer, diabetes, heart disease, Parkinson's, and pneumonia.
2. Evaluate the software's performance in recommending appropriate healthcare professionals for further consultation and diagnosis.

3. Investigate user satisfaction and acceptance of MyMedCare as a tool for self-diagnosis and early intervention.
4. Explore the potential implications of MyMedCare in augmenting traditional healthcare systems, particularly in resource-constrained environments.

Methodology

The research methodology employs a combination of quantitative and qualitative approaches. Quantitative analysis involves testing MyMedCare against a dataset of known cases to measure its sensitivity, specificity, and overall predictive performance. Qualitative methods include user surveys, interviews, and usability testing to gauge user perceptions, preferences, and experiences with the software.

Significance of the Study

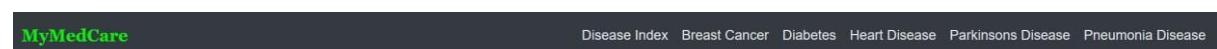
The findings of this research hold significant implications for both healthcare practitioners and technology developers. By elucidating the strengths and limitations of predictive analysis tools like MyMedCare, this study contributes to the ongoing discourse surrounding the integration of artificial intelligence in healthcare delivery. Moreover, insights garnered from user feedback can inform future iterations of MyMedCare, enhancing its usability and effectiveness.

About the interface

This interface contains the self-generated symbol and linked prediction model



It contains the linked streamlit site as

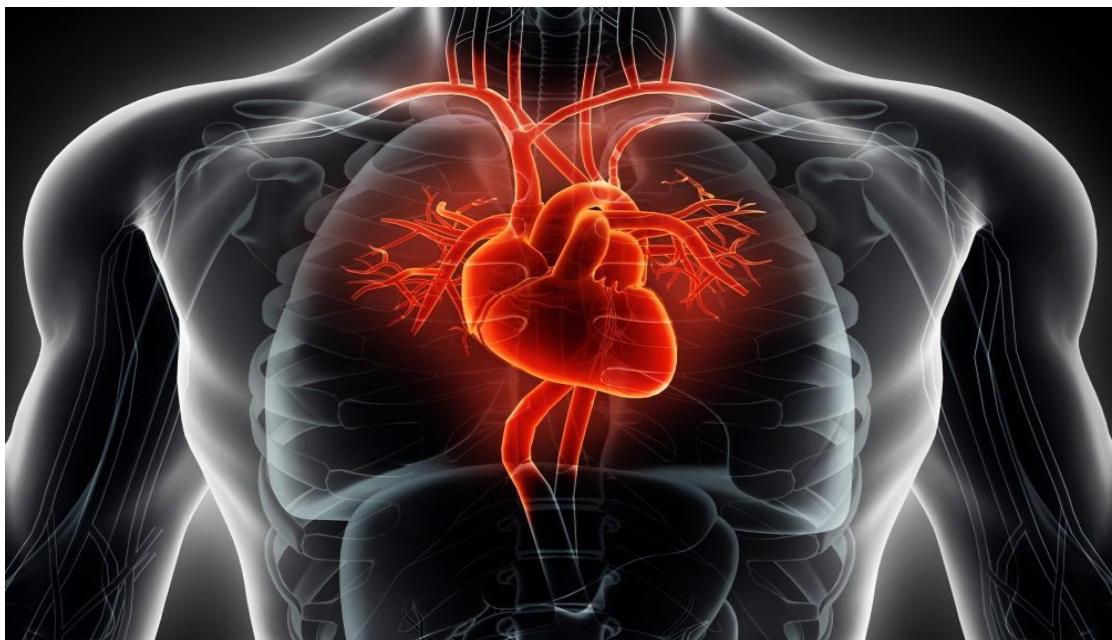


It can redirect you to the streamlit predicted site where you can give your report values and it can then predict the values

About the Diseases

1. Heart Disease

Heart disease, also known as cardiovascular disease, refers to a range of conditions that affect the heart. This comprehensive term encompasses various disorders that affect the heart's structure and function, including coronary artery disease (which can lead to heart attacks), heart rhythm problems (arrhythmias), heart valve diseases, congenital heart defects, and heart failure. Heart disease is a significant global health concern, being a leading cause of death and morbidity worldwide.

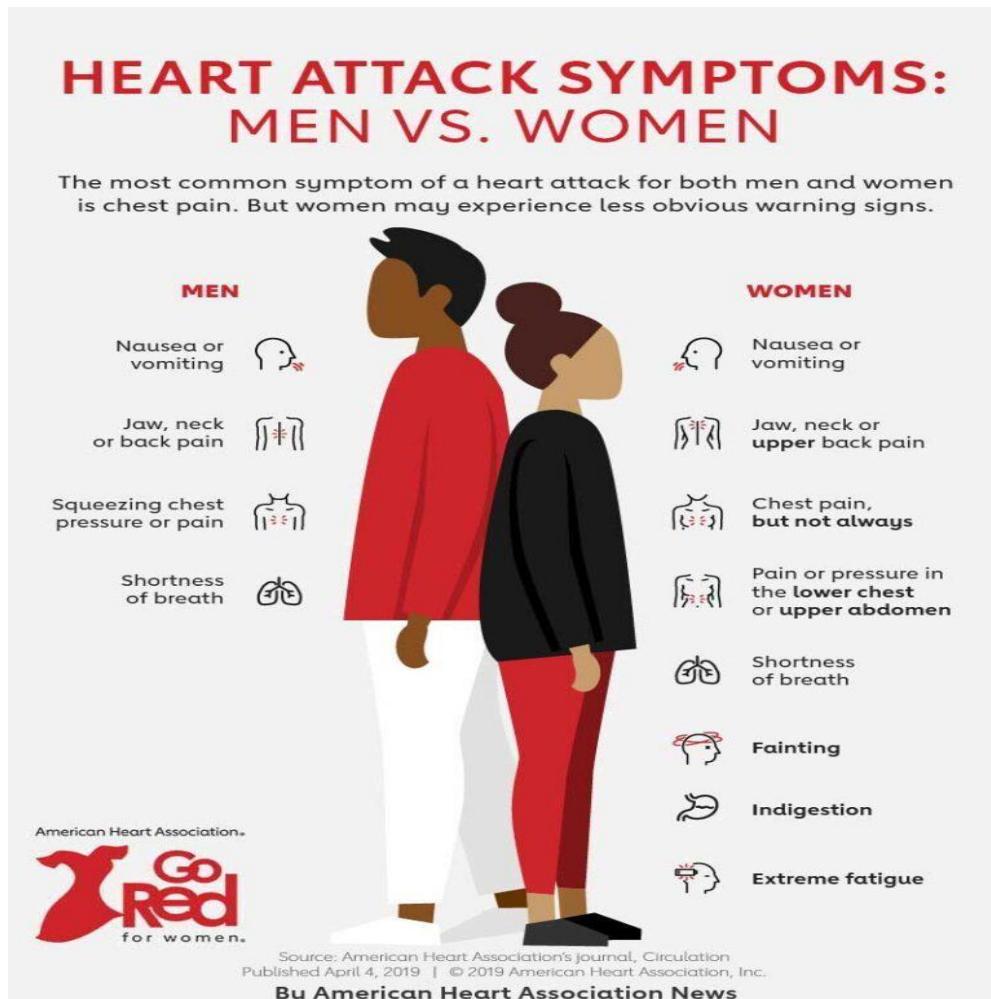


These include unhealthy diet, physical inactivity, smoking, excessive alcohol consumption, obesity, high blood pressure (hypertension), high cholesterol levels (hyperlipidemia), and diabetes

These include age (risk increases with age), family history of heart disease, gender (men are at higher risk at younger ages, while women's risk increases after menopause), and ethnicity (some ethnic groups are at higher risk).

Preventing heart disease involves adopting a healthy lifestyle, including regular exercise, a balanced diet low in saturated fats and sodium, maintaining a healthy weight, avoiding tobacco use, and managing conditions such as high blood pressure, diabetes, and high cholesterol. Screening for risk factors and early detection of heart disease through regular check-ups and diagnostic tests (like blood pressure measurement, lipid profile, electrocardiogram, and echocardiogram) are essential.

Treatment of heart disease depends on the specific condition but may include medications (such as statins, antiplatelet drugs, beta-blockers), lifestyle modifications, surgical interventions (like angioplasty and coronary artery bypass grafting for CAD), cardiac rehabilitation programs, and in severe cases, heart transplantation.



Heart disease is a complex and multifaceted condition that requires a comprehensive approach to prevention, early detection, and management. By addressing risk factors and adopting healthy behaviours, individuals can significantly reduce their risk of developing heart disease and improve their overall cardiovascular health. Ongoing research and advancements in cardiovascular medicine continue to refine treatment strategies and improve outcomes for patients with heart disease.

The root causes of heart disease include atherosclerosis, hypertension, high cholesterol, diabetes, smoking, obesity, genetic factors, and stress.

2. Diabetes

Diabetes is a chronic metabolic disorder characterized by elevated blood glucose (sugar) levels, either due to insufficient insulin production by the pancreas (Type 1 diabetes), ineffective use of insulin by the body (Type 2 diabetes), or a combination of both.



Type 1 Diabetes: This is an autoimmune condition where the immune system attacks and destroys insulin-producing beta cells in the pancreas. As a result, the body is unable to produce enough insulin to regulate blood sugar levels. Type 1 diabetes typically develops in childhood or adolescence, and affected individuals require lifelong insulin therapy.

Type 2 Diabetes: This is the most common form of diabetes, accounting for about 90% of cases. In Type 2 diabetes, the body either becomes resistant to the effects of insulin or doesn't produce enough insulin to maintain normal blood glucose levels. This type of diabetes is often associated with obesity, physical inactivity, and genetic predisposition. Type 2 diabetes can develop at any age, although it is more common in adults.

Gestational Diabetes: This type of diabetes occurs during pregnancy and usually resolves after childbirth. However, women who develop gestational diabetes have an increased risk of developing Type 2 diabetes later in life.

Family history and genetics play a significant role in the development of Type 1 diabetes. In Type 2 diabetes, genetics can influence insulin resistance and beta cell dysfunction.

Excess body weight, especially abdominal obesity, and lack of physical activity contribute to insulin resistance, a key factor in Type 2 diabetes development.

DIABETES SYMPTOMS



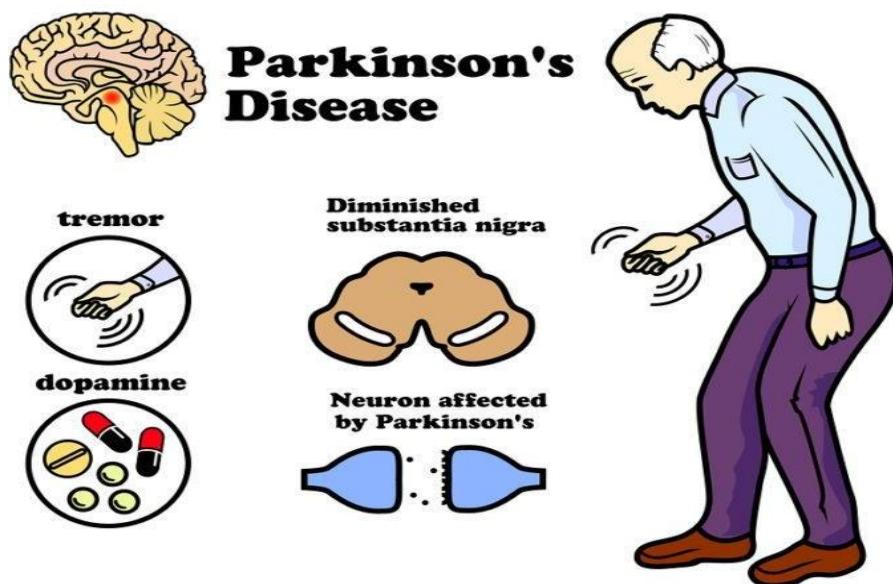
keepvitality.com

In Type 2 diabetes, the body's cells become resistant to insulin, preventing glucose from entering cells and leading to elevated blood sugar levels. In Type 1 diabetes, the pancreas fails to produce insulin due to autoimmune destruction of beta cells. The risk of Type 2 diabetes increases with age, and certain ethnic groups (e.g., Hispanic, African American, Native American, Asian American) are at higher risk.

Diabetes is a complex and chronic condition that requires lifelong management to prevent complications and maintain quality of life. By adopting a proactive approach through healthy lifestyle choices, regular medical care, and adherence to treatment regimens, individuals with diabetes can lead fulfilling and healthy lives while reducing the risk of long-term complications.

3. Parkinson's Disease

Parkinson's disease is a progressive neurodegenerative disorder that primarily affects movement. It develops gradually over time as certain nerve cells in the brain (specifically in the substantia nigra region) become damaged or die, leading to a deficiency of a neurotransmitter called dopamine. Dopamine is crucial for coordinating smooth and controlled muscle movements.



Tremors: Typically starting in the hands or fingers, tremors (shaking) often occur at rest and may progress to affect other parts of the body.

Bradykinesia: This refers to slowness of movement, including difficulty initiating movement, decreased facial expressions (referred to as "masked facies"), and reduced arm swing when walking.

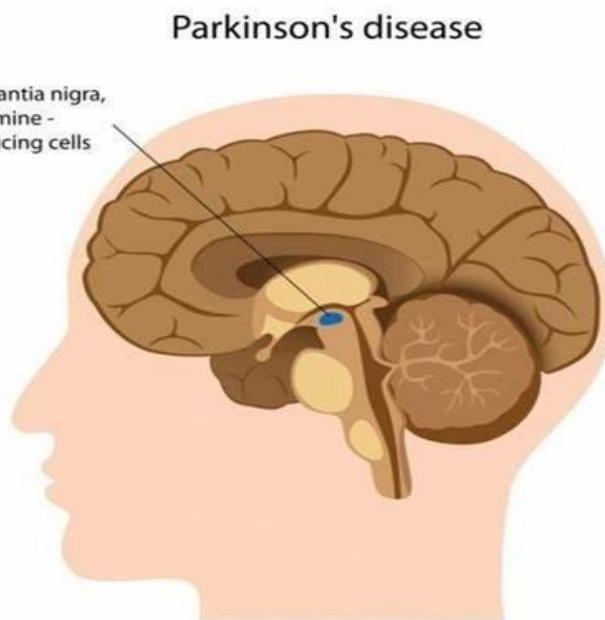
Muscle Rigidity: Stiffness in the limbs and trunk, which can lead to pain and difficulty with movement.

Postural Instability: Impaired balance and coordination, making individuals more prone to falls.

Other Symptoms: Parkinson's disease can also cause non-motor symptoms, including:

- Cognitive changes (e.g., difficulty with memory and concentration)
- Sleep disturbances (e.g., insomnia, REM sleep behavior disorder)
- Mood disorders (e.g., depression, anxiety)
- Loss of sense of smell (anosmia)
- Speech and swallowing difficulties
- Constipation and urinary problems

Parkinson's disease typically develops in people over the age of 60, although younger individuals can also be affected (referred to as young-onset Parkinson's). While most cases of Parkinson's disease are sporadic, certain genetic mutations can increase the risk of developing the condition. Exposure to certain toxins or environmental factors may play a role in the development of Parkinson's disease.



Diagnosing Parkinson's disease is primarily based on clinical symptoms and neurological examination. There are no specific tests to definitively diagnose Parkinson's disease, but neuroimaging (such as MRI or DaTscan) may be used to rule out other conditions with similar symptoms.

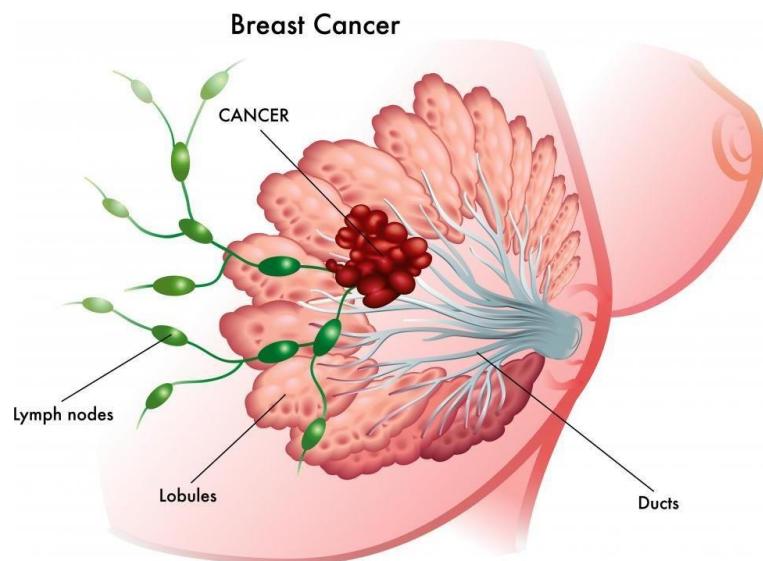
The mainstay of treatment for Parkinson's disease involves medications that help increase dopamine levels in the brain or mimic its effects. These include levodopa, dopamine agonists, MAO-B inhibitors, and COMT inhibitors.

For individuals with advanced Parkinson's disease not adequately controlled with medication, DBS may be considered. This involves implanting electrodes in specific areas of the brain to modulate abnormal neural activity.

Exercises and physical therapy can help improve mobility, flexibility, and balance.

4. Breast Cancer

Breast cancer is a type of cancer that develops in the cells of the breast. It is the most common cancer in women worldwide, but it can also occur in men, although much less frequently. Breast cancer forms when abnormal cells in the breast grow uncontrollably and can invade surrounding tissues or spread to other parts of the body.



Being a woman and age are the primary risk factors for breast cancer, although men can also develop breast cancer.

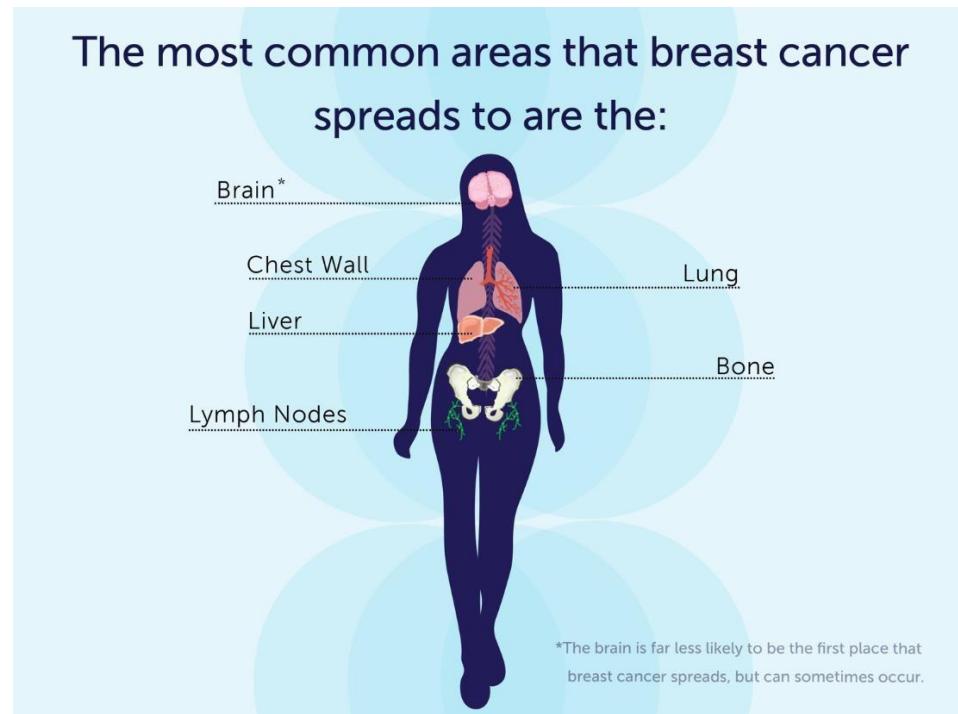
Having close relatives (especially first-degree relatives) with breast cancer increases the risk. Inherited genetic mutations like BRCA1 and BRCA2 also contribute to a higher risk of breast cancer.

Prolonged exposure to estrogen (e.g., early menstruation, late menopause, hormone replacement therapy) increases the risk.

Having had breast cancer in one breast increases the risk of developing cancer in the other breast. Certain benign breast conditions (e.g., atypical hyperplasia) also increase risk.

Factors like obesity, physical inactivity, excessive alcohol consumption, and smoking may increase the risk of breast cancer.

The most common symptom is a painless lump or thickening in the breast or underarm area. Any unexplained changes in breast appearance should be evaluated. Redness, dimpling, or puckering of the breast skin. Changes in nipple appearance (e.g., retraction) or discharge, especially if spontaneous and bloody.



Regular BSE can help detect changes in the breast early.

Women should undergo CBE by a healthcare professional periodically.

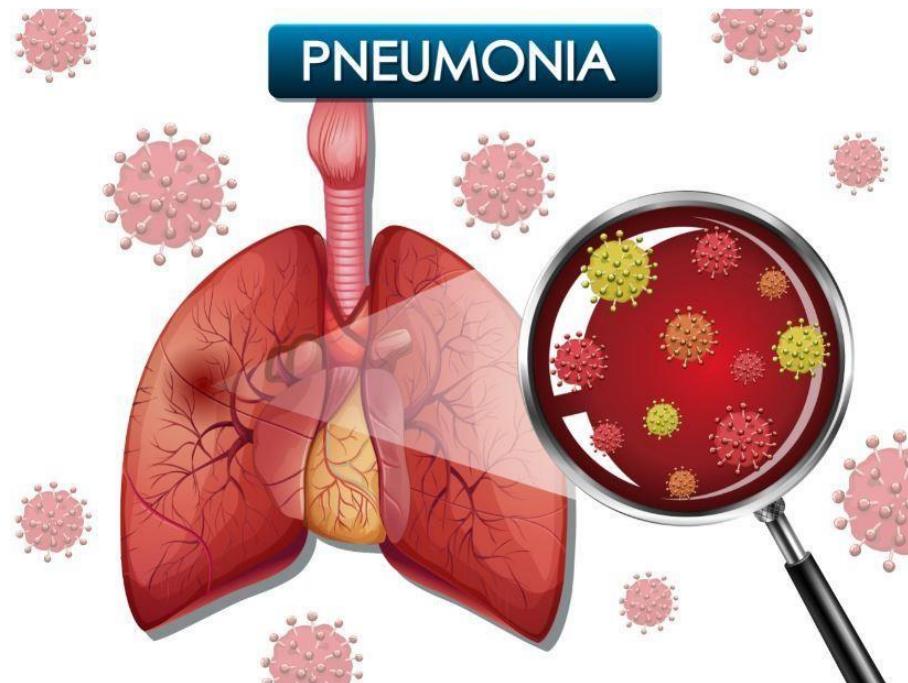
Women should follow recommended guidelines for mammographic screening based on age and risk factors.

Maintaining a healthy weight, being physically active, limiting alcohol consumption, and avoiding smoking can help reduce the risk of breast cancer.

Breast cancer is a serious condition that affects millions of individuals worldwide. Early detection through screening and awareness of risk factors are key to improving outcomes. Advances in treatment and research continue to enhance our understanding of breast cancer and improve survival rates for those affected by this disease.

5. Pneumonia Disease

Pneumonia is a respiratory infection that inflames the air sacs in one or both lungs, causing them to fill with fluid or pus, leading to symptoms such as cough, fever, and difficulty breathing. Pneumonia can be caused by various infectious agents, including bacteria, viruses, fungi, or parasites. It ranges in severity from mild to life-threatening and is a leading cause of illness and death worldwide, particularly among young children, the elderly, and individuals with weakened immune systems.



The most common cause of bacterial pneumonia is *Streptococcus pneumoniae* (pneumococcus). Other bacteria, such as *Haemophilus influenzae* and *Mycoplasma pneumoniae*, can also cause bacterial pneumonia. Influenza (flu) viruses, respiratory syncytial virus (RSV), and adenoviruses are common causes of viral pneumonia. Fungal infections, such as *Pneumocystis jirovecii* (associated with HIV/AIDS) and *Histoplasma capsulatum* (common in certain geographic areas), can cause fungal pneumonia, especially in individuals with weakened immune systems. This occurs when food, liquids, saliva, or vomit is inhaled into the lungs, leading to inflammation and infection.

- Cough: Often producing mucus (may be green or yellow).
- Fever: High fever with chills.
- Shortness of Breath: Difficulty breathing, especially with exertion.
- Chest Pain: Sharp or stabbing pain when breathing deeply or coughing.
- Fatigue: Feeling unusually tired or weak.
- Nausea, Vomiting, or Diarrhea: Especially common in children.

- Age: Very young children and the elderly are at higher risk.
- Weakened Immune System: Due to conditions like HIV/AIDS, cancer, or organ transplantation.
- Chronic Illnesses: Such as chronic obstructive pulmonary disease (COPD), asthma, or heart disease.
- Smoking: Weakened lung defences.
- Hospitalization: Especially for those on ventilators or with extended bed rest.

Symptoms of Pneumonia



Including symptoms, risk factors, and recent infections. To confirm the presence and extent of lung inflammation. To identify the infectious agent causing pneumonia. To identify the specific bacteria causing bacterial pneumonia. Vaccines against Streptococcus pneumoniae (pneumococcus) and influenza (flu) can prevent pneumonia.

Pneumonia is a common and serious respiratory infection that can affect individuals of all ages, with higher risks for certain populations. Early recognition of symptoms, prompt medical evaluation, appropriate treatment, and preventive measures such as vaccination are essential for reducing the burden of pneumonia and improving outcomes for affected individuals.

In an era of advancing technology and personalized healthcare, MyMedCare stands as your trusted companion in managing and understanding critical health conditions. From predicting heart disease and diabetes risks to offering insights on pneumonia, Parkinson's disease, breast health, and more, MyMedCare leverages the power of data science and medical expertise to empower individuals with proactive health insights.

Harnessing state-of-the-art machine learning algorithms, MyMedCare provides personalized predictions and recommendations based on individual health data. Whether you're concerned about your cardiovascular health, seeking early detection of diabetes, or exploring preventative measures for Parkinson's disease or breast cancer, MyMedCare is here to assist you every step of the way.

By combining cutting-edge technology with a user-friendly interface, MyMedCare transforms complex medical data into actionable insights. Our mission is to enable informed decision-making, promote early intervention, and ultimately improve health outcomes for everyone.

Overview.

MyMedCare empowers individuals to take control of their health by providing actionable insights and personalized recommendations. Our user-friendly interface and intuitive design make it easy for users to access and interpret health information, fostering a proactive approach to health management.

Join MyMedCare today to embark on a journey towards better health and well-being. Together, let's leverage the power of data-driven healthcare to optimize health outcomes and enhance quality of life. Your health, your care, our priority.



Chapter 2 : Literature Survey

Tools and Techniques

1. Deep Learning

In the realm of personalized health management, MyMedCare harnesses the transformative capabilities of deep learning, a subset of artificial intelligence focused on training complex neural networks to extract patterns and insights from large datasets. Deep learning techniques play a pivotal role in advancing predictive models and enabling more accurate assessments of critical health conditions within the MyMedCare platform.

Deep learning models autonomously learn hierarchical representations of features from raw data, eliminating the need for manual feature engineering. This allows MyMedCare to process diverse inputs, such as medical images, clinical measurements, and genetic profiles, effectively.

Deep learning excels in handling complex, high-dimensional datasets, resulting in improved predictive accuracy for health outcomes. By continuously learning from data, MyMedCare's deep learning models adapt to new information and evolving patterns, refining predictions over time.

Deep learning enables MyMedCare to generate personalized health recommendations tailored to individual users. By analyzing vast amounts of historical and real-time data, the platform can suggest optimized preventive measures, treatment options, and lifestyle interventions.

MyMedCare's deep learning models are designed for continuous learning and adaptation. As new research and data become available, the platform evolves to integrate the latest advancements in health science, ensuring up-to-date and effective health management strategies.

Deep learning models, particularly convolutional neural networks (CNNs), excel in analyzing medical imaging data such as chest X-rays and CT scans. Pneumonia detection often involves interpreting these images to identify signs of infection, such as consolidations, infiltrates, or opacities in the lungs.

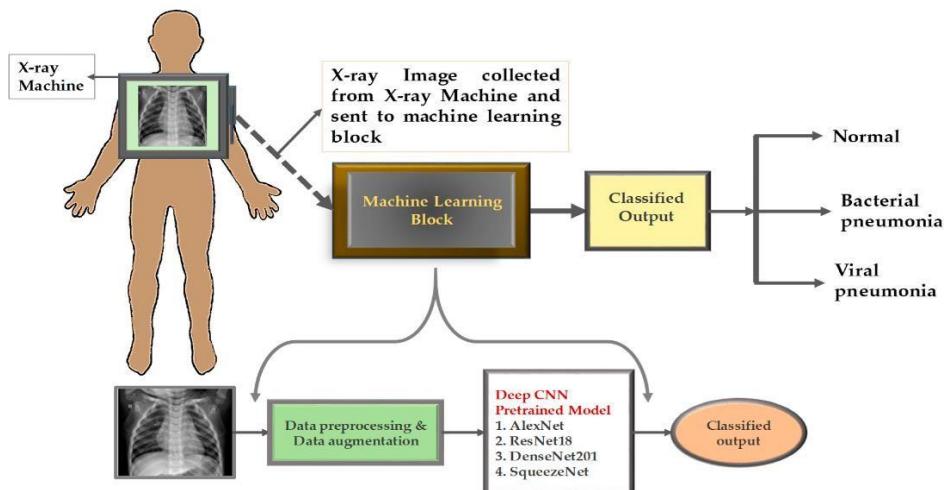
Deep learning models require large annotated datasets of chest X-ray images labeled with pneumonia or non-pneumonia (normal) classes.

Standardizing image sizes, normalization, and augmentation techniques to improve model generalization.

Utilizing annotated data to train deep learning models to identify patterns indicative of pneumonia.

Evaluating model performance on validation datasets to ensure robustness and generalizability.

- **CNN** : Convolutional Neural Networks (CNNs) have been instrumental in advancing the field of medical image analysis, particularly in the detection and diagnosis of pneumonia from chest X-ray images. CNNs require a large dataset of chest X-ray images labeled with binary classifications (pneumonia vs. normal) for training purposes. Techniques such as rotation, flipping, scaling, and contrast adjustment are applied to increase dataset diversity and improve model generalization. These layers use learnable filters to extract spatial features from input chest X-ray images. ReLU (Rectified Linear Unit) is commonly used to introduce non-linearity. These layers combine extracted features for final classification. Metrics such as accuracy, precision, recall, and F1-score are used to evaluate model performance. Helps identify false positives and false negatives, which are crucial in medical diagnosis.
Trained CNN models are deployed as part of diagnostic systems or integrated into healthcare workflows to assist radiologists and clinicians in pneumonia diagnosis.



Some systems provide real-time predictions on new chest X-ray images for immediate decision-making. Developed by Stanford researchers, this CNN model demonstrated high accuracy in pneumonia detection using chest X-ray images.

2. Machine Learning

Machine Learning is a branch of artificial intelligence (AI) that enables systems to learn and improve from experience without being explicitly programmed. It involves the development of algorithms and statistical models that allow computers to perform tasks by learning from data rather than following specific instructions.

ML algorithms learn patterns and relationships from data to make predictions or decisions.

Models are trained using labeled data (supervised learning) or unlabeled data (unsupervised learning) to make predictions on new data (inference). ML models generalize from training data to make accurate predictions on unseen data, avoiding overfitting (memorizing training data) and underfitting (failing to capture patterns).

Supervised Learning : Uses labeled data to train models to make predictions (classification) or estimate continuous values (regression).

Unsupervised Learning : Learns patterns and structures from unlabeled data, including clustering (grouping similar data) and dimensionality reduction.

Reinforcement Learning : Teaches agents to make decisions in an environment to maximize cumulative reward, often used in game playing and robotics.

ML enables accurate image classification, object detection, and speech recognition, powering applications like facial recognition and virtual assistants. ML models process and understand human language, enabling sentiment analysis, language translation, chatbots, and text summarization. ML algorithms analyze user preferences and behavior to make personalized recommendations, seen in e-commerce, streaming platforms, and social media. ML predicts future outcomes based on historical data, used in finance (stock price prediction), healthcare (disease prognosis), and marketing (customer behavior). ML identifies unusual patterns or outliers in data, critical for fraud detection, network security, and predictive maintenance. ML enables self-driving cars to perceive and navigate environments using sensors and real-time data analysis. ML aids in disease diagnosis, medical imaging analysis, drug discovery, and personalized treatment planning based on patient data. ML assesses credit risk, detects fraudulent transactions, optimizes trading strategies, and automates customer service in banking and insurance.

Automates repetitive tasks and decision-making processes, improving efficiency and reducing human error. Creates personalized experiences and recommendations based on individual preferences and behavior. Scales to process large volumes of data,

enabling real-time insights and decision-making. Adapts to changing data patterns and environments, continuously improving performance over time.

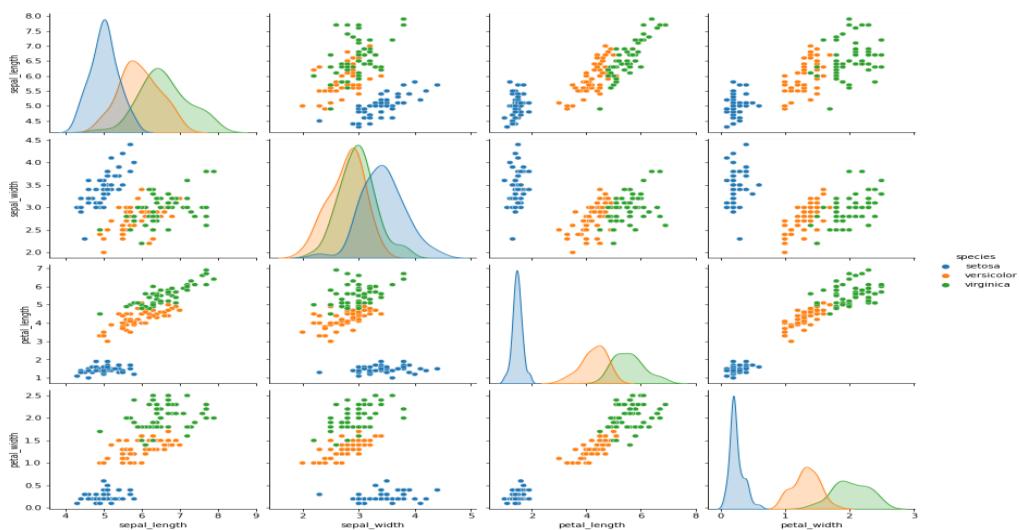
- **Scikit – learn (sklearn)**

Scikit-learn, often abbreviated as sklearn, is a popular open-source machine learning library for Python. It provides simple and efficient tools for data mining and analysis, built on NumPy, SciPy, and matplotlib. Scikit-learn is designed to work seamlessly with other Python libraries and frameworks like pandas and TensorFlow, making it a versatile choice for machine learning tasks. Scikit-learn offers a consistent API for various machine learning tasks, making it easy to experiment with different algorithms and techniques. It provides implementations of a variety of supervised and unsupervised learning algorithms, including classification, regression, clustering, dimensionality reduction, and more. Scikit-learn emphasizes simplicity and ease of use, with straightforward interfaces and sensible default parameters. Scikit-learn seamlessly integrates with NumPy arrays and pandas DataFrames, making it compatible with standard data manipulation workflows in Python. It includes utilities for preprocessing data, such as scaling, normalization, encoding categorical variables, and handling missing values. Scikit-learn provides tools for model selection and evaluation, including cross-validation, hyperparameter tuning, and performance metrics like accuracy, precision, recall, and F1-score. It supports the construction of machine learning pipelines, allowing you to chain together preprocessing steps and model training for streamlined workflows. Scikit-learn's official documentation (<https://scikit-learn.org/stable/>) provides comprehensive guides, tutorials, and examples to help you get started with different machine learning tasks using the library. Explore the documentation and experiment with various algorithms and techniques supported by scikit-learn to tackle your machine learning projects efficiently.

- **Bivariate Data Analysis**

Bivariate data analysis focuses on understanding the relationship between two variables simultaneously. This type of analysis is fundamental in statistics and data science, as it allows us to explore how changes in one variable affect changes in another. A scatter plot is a graphical representation that displays the relationship between two continuous variables. Scatter plots help visualize patterns, trends, and associations between variables. The shape, direction, and density of points in a scatter plot provide insights into the strength and nature of the relationship between variables

(e.g., positive, negative, or no correlation). The correlation coefficient (e.g., Pearson's correlation coefficient) quantifies the strength and direction of the linear relationship between two continuous variables. The correlation coefficient ranges from -1 (perfect negative correlation) to 1 (perfect positive correlation), with 0 indicating no linear correlation. A correlation value close to -1 or 1 indicates a strong linear relationship, while values close to 0 suggest weak or no linear relationship. Covariance measures the degree to which two variables change together (i.e., how much one variable varies with another). Positive covariance indicates that the variables tend to increase or decrease together, while negative covariance suggests that one variable increases as the other decreases.

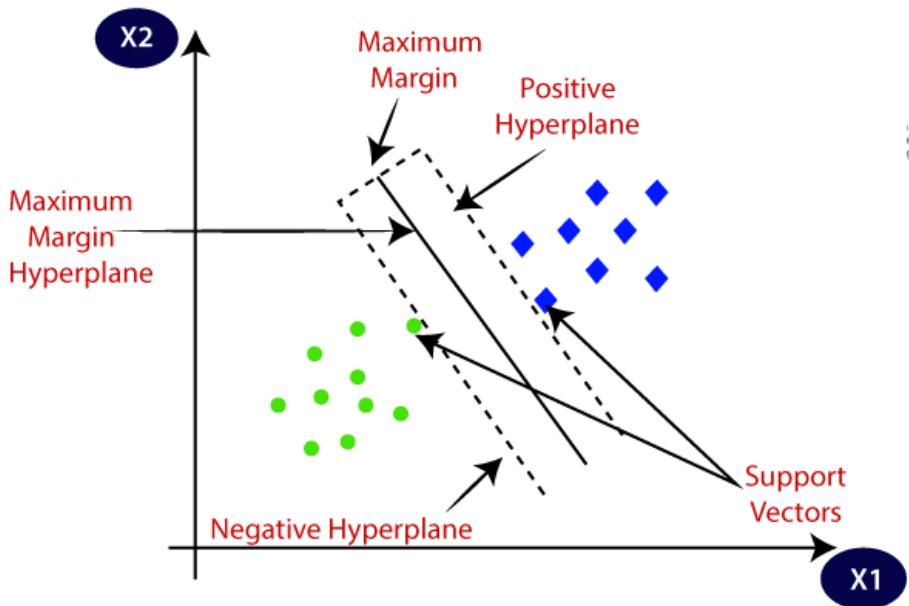


Covariance's value is dependent on the units of the variables, making it less interpretable than correlation coefficient for comparing relationships. A contingency table (also known as a cross-tabulation table) summarizes the distribution of two categorical variables. This statistical test assesses the independence of two categorical variables based on observed frequencies in a contingency table. A significant chi-square test result suggests that the two categorical variables are associated or dependent. Examines the relationship between two variables, where one variable (predictor) is used to predict another variable (outcome) through a linear model. Extends simple linear regression to analyze the relationship between one dependent variable and multiple independent variables. Regression analysis helps quantify and predict the impact of one variable on another, accounting for other potential confounding variables.

Visualize the correlation matrix between multiple variables, providing a comprehensive view of pairwise relationships. Display correlation coefficients between all pairs of variables, facilitating pattern identification and feature selection.

- **Support Vector Machine**

Support Vector Machine (SVM) is a powerful supervised learning algorithm used for classification and regression tasks. SVM is particularly effective in high-dimensional spaces and is capable of finding complex decision boundaries by maximizing the margin between different classes of data points. SVM aims to find the optimal hyperplane that best separates two classes in the feature space. SVM maximizes the margin, which is the distance between the hyperplane and the nearest data points (support vectors) of each class. The decision boundary is determined by the hyperplane that maximizes the margin and maintains the largest separation between classes. SVM can handle non-linear decision boundaries by mapping input features into a higher-dimensional space using kernel functions (e.g., polynomial kernel, radial basis function (RBF) kernel). The kernel trick allows SVM to compute the decision boundary in the higher-dimensional space efficiently without explicitly transforming the input features.



Support vectors are the data points that lie closest to the decision boundary. They are critical for defining the optimal hyperplane in SVM. Only the support vectors influence the position and orientation of the decision boundary, making SVM memory-efficient and effective in high-dimensional spaces. Controls the trade-off

between maximizing the margin and minimizing the classification error on the training data. SVM uses regularization to prevent overfitting by penalizing large coefficients in the decision function. Constructs multiple binary classifiers for each pair of classes and combines their predictions. Constructs multiple binary classifiers for each pair of classes and combines their predictions. SVM performs well even when the number of dimensions is greater than the number of samples. SVM maximizes the margin, making it less prone to overfitting compared to other algorithms like decision trees. SVM supports different kernel functions, allowing flexibility in modeling complex decision boundaries.

- **Confusion Matrix**

A confusion matrix (or error matrix) is a table used to evaluate the performance of a classification model by summarizing the predicted versus actual class labels. It is a fundamental tool for assessing the accuracy, precision, recall, and F1-score of a classifier across different classes. The confusion matrix is especially useful in binary and multi-class classification problems.

		PREDICTED	
		Positive	Negative
ACTUAL	Positive	TRUE POSITIVE	FALSE NEGATIVE
	Negative	FALSE POSITIVE	TRUE NEGATIVE

dataaspirant.com

True Positive (TP): Predicted as Positive (P) and actually Positive (P).

True Negative (TN): Predicted as Negative (N) and actually Negative (N).

False Positive (FP): Predicted as Positive (P) but actually Negative (N) (Type I error).

False Negative (FN): Predicted as Negative (N) but actually Positive (P) (Type II error).

Confusion Matrix Structure:

	Predicted Negative (N)	Predicted Positive (P)
Actual Negative (N)	True Negative (TN)	False Positive (FP)
Actual Positive (P)	False Negative (FN)	True Positive (TP)

Evaluation Metrics Derived from Confusion Matrix:1. **Accuracy:**

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

2. **Precision** (Positive Predictive Value):

$$\text{Precision} = \frac{TP}{TP + FP}$$

3. **Recall** (Sensitivity, True Positive Rate):

$$\text{Recall} = \frac{TP}{TP + FN}$$

4. **F1-Score** (Harmonic Mean of Precision and Recall):

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Logistic Regression**

Logistic Regression is a statistical method used for binary and multi-class classification tasks. Despite its name, logistic regression is a classification algorithm rather than a regression algorithm. It models the probability of a binary outcome based on one or more predictor (independent) variables. Logistic regression models the probability $P(y=1|x)$ given predictor variables x .

The logistic function (sigmoid) is used to transform the linear combination of predictors into a probability score between 0 and 1.

$$P(y = 1 | \mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^\top \mathbf{x}-b}}$$

For binary logistic regression, the decision boundary is a hyperplane separating classes based on predicted probabilities. In multi-class logistic regression, the decision boundary is determined by comparing probabilities across multiple classes. Minimize the logistic loss (binary) or cross-entropy loss (multi-class) using optimization algorithms like Gradient Descent. L1 (Lasso) or L2 (Ridge) regularization can be applied to prevent overfitting.

Accuracy: Measures overall correctness of the model predictions.

Precision: Proportion of true positive predictions among positive class predictions.

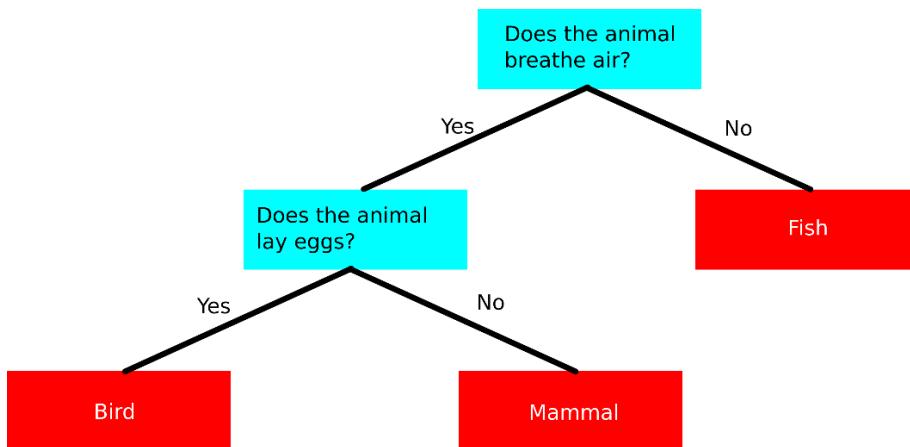
Recall (Sensitivity): Proportion of actual positives that were correctly predicted.

F1-Score: Harmonic mean of precision and recall, useful for imbalanced datasets.

Logistic Regression is widely used in various applications, including healthcare (disease prediction), finance (credit risk assessment), and natural language processing (text classification), due to its simplicity, interpretability, and effectiveness in modeling categorical outcomes based on predictor variables.

- **Decision Tree Classifier**

A Decision Tree Classifier is a popular supervised learning algorithm used for both classification and regression tasks. It builds a tree-like structure where each internal node represents a decision based on a feature, and each leaf node represents the outcome (class label or value). Decision trees are intuitive and easy to interpret, making them suitable for a wide range of applications.



Nodes: Represent decisions based on features.

Edges: Connect nodes and represent possible outcomes or decisions.

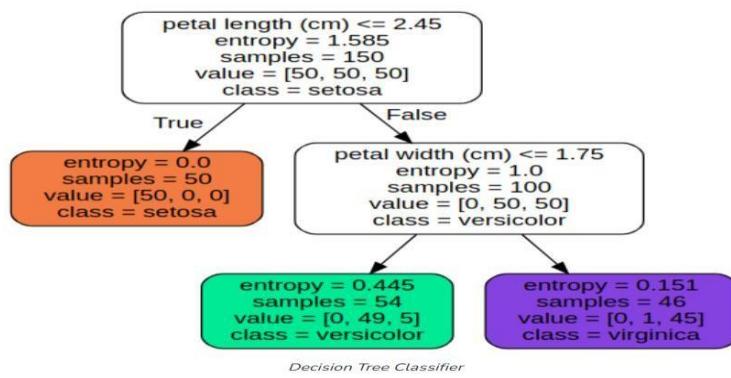
Root Node: Represents the starting point of the decision-making process.

Internal Nodes: Correspond to features and conditions used for splitting data.

Leaf Nodes: Represent class labels or regression values.

Gini Impurity (for Classification): Measures the impurity or disorder of a set of samples. The goal is to minimize the impurity at each split.

Output:



Entropy: Measures the level of randomness or uncertainty in a set of samples.

Information Gain: Quantifies the reduction in entropy or impurity after a split. The decision tree selects the feature that maximizes information gain.

Recursive Binary Splitting: The tree is grown recursively by splitting the dataset into smaller subsets based on feature conditions.

Stopping Criteria: Tree growth stops when a predefined condition is met (e.g., maximum depth of the tree, minimum number of samples per leaf).

Categorical Features: Decision trees can handle categorical features directly by partitioning based on category values.

Numerical Features: Decision trees select thresholds to split numerical features into discrete intervals.

Interpretability: Easy to visualize and understand, suitable for explaining decision-making processes to stakeholders.

No Data Preprocessing: Decision trees do not require feature scaling or normalization.

Handle Non-linear Relationships: Decision trees can capture complex non-linear relationships between features and target variables.

Pre-Pruning: Stop growing the tree early based on criteria such as maximum depth, minimum samples per leaf, or maximum number of leaf nodes.

Post-Pruning (or Pruning): Trim the tree after it is fully grown to reduce overfitting and improve generalization.

Decision trees are versatile and widely used in various domains, including healthcare (disease diagnosis), finance (credit risk assessment), and marketing (customer segmentation), due to their ability to handle both categorical and numerical data, interpretability, and effectiveness in capturing non-linear relationships in data. However, care must be taken to prevent overfitting, especially with deep and complex trees, by using appropriate pruning techniques and regularization.

- **Guassian Naïve Bayes (GaussianNB)**

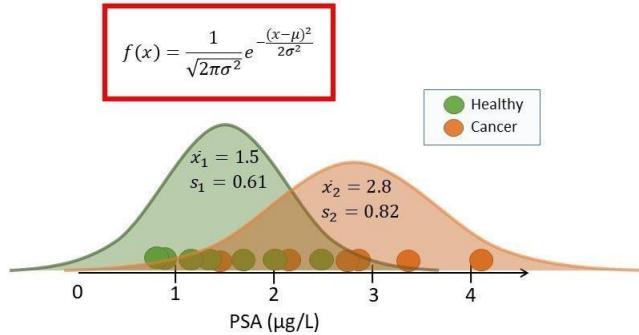
Gaussian Naive Bayes (GaussianNB) is a popular classification algorithm based on Bayes' Theorem with an assumption of independence between features. Despite its simplicity, GaussianNB is effective for many classification tasks, especially when dealing with continuous numerical data. Calculates the probability of a class C_k given the features x using Bayes' Theorem:

$$P(C_k | \mathbf{x}) = \frac{P(\mathbf{x}|C_k) \times P(C_k)}{P(\mathbf{x})}$$

GaussianNB assumes that the features x_1, x_2, \dots, x_n are conditionally independent given the class label C_k .

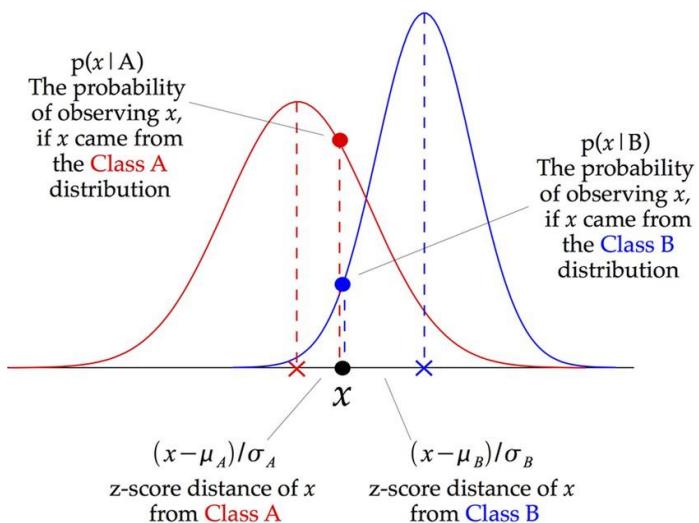
Gaussian Naive Bayes

Status	PSA
Cancer	4.1
Cancer	3.4
Cancer	2.9
Cancer	2.8
Cancer	2.7
Cancer	2.1
Cancer	1.6
Healthy	2.5
Healthy	2.0
Healthy	1.7
Healthy	1.4
Healthy	1.2
Healthy	0.9
Healthy	0.8



TileStats

Each feature x_i is assumed to follow a Gaussian (normal) distribution. GaussianNB estimates the parameters (mean and variance) of the Gaussian distribution for each class based on the training data. Prior probability $P(C_k)$ is estimated based on the class distribution in the training set. GaussianNB uses the maximum a posteriori (MAP) decision rule to predict the class label for new instances based on the highest posterior probability.

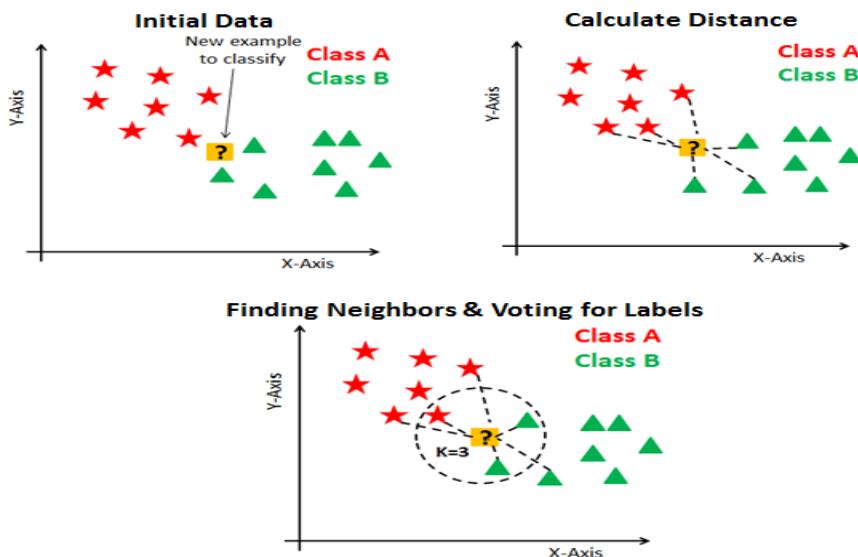


Gaussian Naive Bayes is suitable for various classification tasks, particularly when dealing with continuous numerical features that can be assumed to follow a Gaussian distribution. It is widely used in text classification, spam detection, and medical

diagnosis due to its simplicity and efficiency. However, it may not perform well with highly correlated features or non-Gaussian data distributions, requiring careful evaluation and preprocessing of the input data.

- **K-Nearest Neighbors (KNN) Classifier**

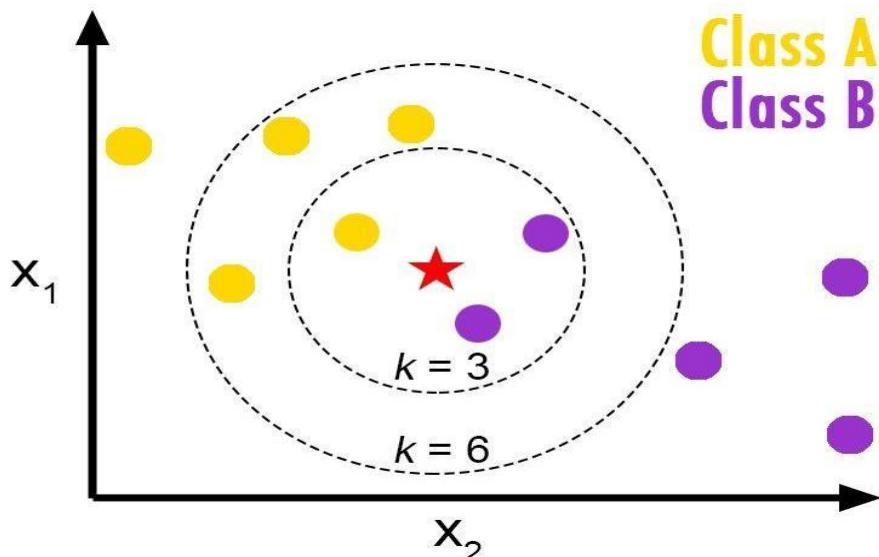
The K-Nearest Neighbors (KNN) classifier is a simple and intuitive supervised learning algorithm used for classification tasks. It is based on the principle that similar data points are likely to belong to the same class. KNN is a non-parametric, lazy learning algorithm that does not make assumptions about the underlying distribution of the data. KNN typically uses Euclidean distance to measure the similarity between data points in feature space. Other distance metrics such as Manhattan distance or cosine similarity can also be used based on the problem domain.



KNN requires specifying the number of nearest neighbors to consider when making predictions. The value of K is a hyperparameter that needs to be tuned based on the dataset and problem.

For classification, KNN predicts the class of a new data point based on the majority class among its K nearest neighbors. Optionally, weights can be assigned to neighboring points based on their distance to the query point, giving more influence to closer neighbors. KNN does not explicitly train a model during the training phase. Instead, it memorizes the training data and makes predictions based on similarities at runtime. The computational cost of KNN increases with the size of the training set since it requires computing distances to all training samples during prediction.

KNN is sensitive to the scale of features due to its reliance on distance-based calculations. It is recommended to scale or normalize features before applying KNN. KNN is straightforward to understand and implement, making it suitable for baseline models and quick prototyping. KNN does not make strong assumptions about the underlying data distribution, making it versatile for different types of datasets.



KNN can be computationally expensive, especially with large datasets and high-dimensional feature spaces. Performance of KNN can degrade in high-dimensional spaces due to increased sparsity and distance calculation complexity. KNN is one of the most basic yet essential classification algorithms in machine learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining, and intrusion detection.

3. HTML / CSS

- **HTML**

HTML (HyperText Markup Language) is the standard markup language used to create the structure and content of web pages. It consists of a series of elements (tags) that define the different parts of a web document. HTML elements are surrounded by angle brackets ($<>$) and usually come in pairs with an opening tag and a closing tag.

<!DOCTYPE html>: Declaration that specifies the HTML version being used (HTML5 in this case).

<html>: Root element that wraps the entire HTML document.

<head>: Contains meta-information about the HTML document (e.g., character set, viewport settings, page title).

<meta>: Provides metadata about the HTML document, such as character encoding (charset) and viewport settings for responsive design.

<title>: Sets the title of the web page, which appears in the browser tab.

<body>: Contains the visible content of the web page, including headers, paragraphs, lists, and other elements.

<header>: Represents introductory content at the top of the page, typically containing headings, navigation links, or logos.

<h1>, <h2>, <h3>, ...: Headings used to define the importance of text content (from most important **<h1>** to least important **<h6>**).

<nav>: Defines navigation links, often presented as a menu.

, , **: ** (unordered list) and **** (ordered list) are used to create lists.

**** (list item) defines each item in a list.

<a>: Anchor element used to create hyperlinks (**Link Text**).

<section>: Represents a thematic grouping of content within the document.

<p>: Defines a paragraph of text.

<div>: Generic container used to group and style content.

Web browsers interpret HTML documents to render web pages visually. HTML tags define the structure and content of the page, while CSS is used to style and layout the elements. Modern web development often involves combining HTML with CSS and JavaScript to create dynamic and interactive websites. HTML is the foundation of web development, providing the structure and semantics that browsers use to display content. By learning HTML, you gain the ability to create static web pages and understand the underlying structure of more complex web applications. To enhance the appearance and behavior of HTML pages, CSS and JavaScript are often used in conjunction with HTML.

- **CSS**

CSS (Cascading Style Sheets) is a language used to describe the presentation and styling of HTML elements on a web page. It allows developers to control the layout, appearance, and behavior of web pages, making them more visually appealing and user-friendly. CSS selectors are used to target HTML elements and apply styles to them. Common selectors include element selectors (e.g., p, h1, div), class selectors (e.g., .classname), and ID selectors (e.g., #idname). CSS styles are applied using declaration blocks {} containing one or more property-value pairs. Multiple CSS rules

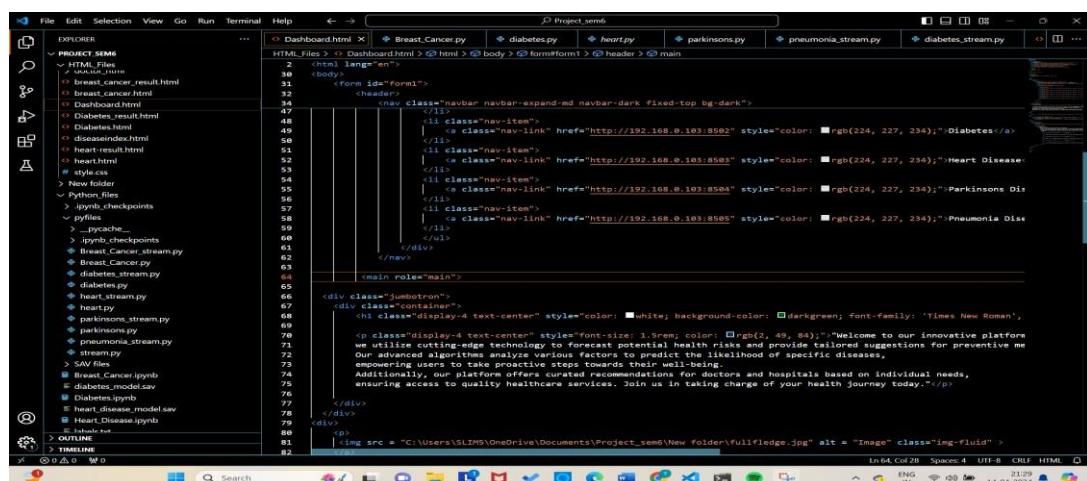
can be combined to style different elements on a web page. The CSS box model describes how elements are rendered in the browser, including content area, padding, border, and margin. Understanding the box model is crucial for controlling the spacing and layout of elements. CSS supports various units of measurement for specifying sizes and distances, including pixels (px), percentages (%), em units (em), rem units (rem), viewport units (vw, vh), etc. CSS styles are typically written in separate .css files and linked to HTML documents using <link> tags in the <head> section of the HTML document. CSS styles can also be included directly within HTML documents using <style> tags. Browsers apply CSS styles to HTML elements based on the selectors specified in the CSS rules.

CSS frameworks like Bootstrap, Foundation, and Materialize provide pre-built styles and components to create responsive and visually appealing websites.

CSS libraries like SASS (Syntactically Awesome Style Sheets) and LESS (Leaner Style Sheets) extend the capabilities of CSS by adding features like variables, mixins, and nested rules.

4. VS CODE

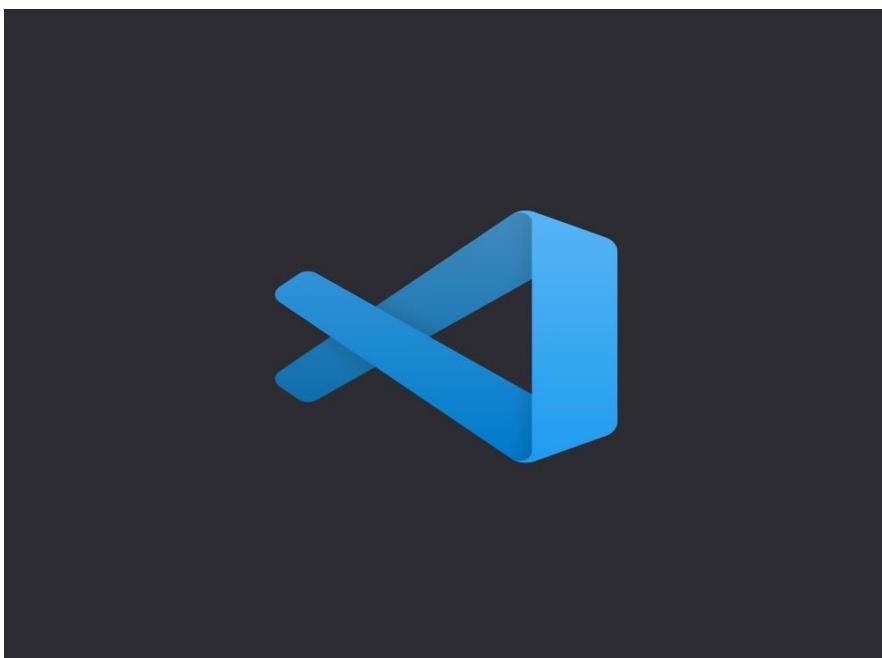
Visual Studio Code (VS Code) is a popular and versatile source code editor developed by Microsoft. It is widely used by developers for coding, debugging, and version control tasks across various programming languages and frameworks. VS Code is known for its lightweight yet powerful features, extensive customization options, and vibrant ecosystem of extensions. Available for Windows, macOS, and Linux operating systems, providing a consistent development experience across platforms. Clean and customizable interface with a sidebar for file navigation, integrated terminal, and activity bar for quick access to tools and commands.



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a folder named "Project_sem6" containing several Python files (Breast_Cancer.py, diabetes.py, heart.py, parkinsons.py, pneumonia_stream.py, diabetes_stream.py) and a dashboard.html file.
- Editor:** Displays the content of the dashboard.html file. The HTML includes a header with a logo and navigation links for Diabetes, Heart Disease, Parkinsons Disease, and Pneumonia Disease. Below the header is a main content area with a jumbotron containing a welcome message about the platform's advanced algorithms and preventive measures.
- Terminal:** Shows the command "ls" being run, displaying the contents of the current directory.
- Activity Bar:** Includes icons for file operations, search, and other development tools.
- Bottom Status Bar:** Shows the file path ("C:\Users\SLIMS\OneDrive\Documents\Project_sem6\New Folder\fullfledge.jpg"), encoding ("UTF-8"), and other system information.

Built-in support for a wide range of programming languages including JavaScript, TypeScript, Python, Java, C/C++, HTML/CSS, and more. Features like syntax highlighting, code completion, and IntelliSense enhance productivity and code readability. Extensive ecosystem of extensions available through the Visual Studio Code Marketplace. Extensions provide additional features such as debugging tools, language support, themes, version control integrations, and productivity enhancements. Built-in terminal allows developers to run command-line tools and scripts directly within the editor.



Supports multiple terminal instances and customization options. Seamless integration with Git and other version control systems (e.g., GitHub, Azure DevOps). Visual Studio Code provides built-in Git commands, version control views, and source control features. Built-in debugging support for various programming languages with breakpoints, watch variables, call stacks, and interactive debugging tools. Supports debugging Node.js applications, web apps, and other environments. Visual Studio Code is widely used by developers for web development, application development, data science, and more. Its rich feature set, active community support, and continuous improvement make it a popular choice for developers across different domains and skill levels. Whether you're a beginner or an experienced developer, VS Code provides the tools and flexibility needed to efficiently write, debug, and manage code projects.

5. Streamlit

Streamlit is an open-source Python library that makes it easy to create and share web applications for machine learning and data science projects. It allows you to build interactive and responsive web interfaces directly from your Python scripts, without requiring knowledge of HTML, CSS, or JavaScript. Streamlit is particularly useful for quickly prototyping and deploying data-driven applications.

Streamlit is designed for simplicity and ease of use. You can create powerful web apps with just a few lines of Python code.

Streamlit provides a wide range of interactive widgets (e.g., sliders, dropdowns, buttons) that allow users to interact with your machine learning models and visualizations.



Streamlit

Streamlit automatically updates your app in real-time as you modify your Python script, making the development process seamless.

You can easily integrate popular data science libraries like Pandas, Matplotlib, and TensorFlow into your Streamlit app.

Streamlit apps can be deployed locally or on cloud platforms such as Heroku, AWS, or Google Cloud.

Use Streamlit to build intuitive and interactive web applications for your machine learning and data science projects, and easily share your work with others.

Experiment with different widgets, visualizations, and machine learning models to create impactful applications with Streamlit.

Chapter 3 : Mathematical Background / logical thinking/ fundamental concepts

1. Heart Disease Prediction:

Heart disease prediction using machine learning involves applying statistical and computational methods to a dataset containing features that may correlate with heart disease risk. The goal is to build a model that can accurately predict whether a person has a high or low risk of heart disease based on these features.

Heart disease prediction typically uses a dataset with various attributes like age, sex, cholesterol levels, blood pressure, etc.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
2	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
3	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
4	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
5	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
6	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
7	57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
8	56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
9	44	1	1	120	263	0	1	173	0	0	2	0	3	1
10	52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
11	57	1	2	150	168	0	1	174	0	1.6	2	0	2	1
12	54	1	0	140	239	0	1	160	0	1.2	2	0	2	1
13	48	0	2	130	275	0	1	139	0	0.2	2	0	2	1
14	49	1	1	130	266	0	1	171	0	0.6	2	0	2	1
15	64	1	3	110	211	0	0	144	1	1.8	1	0	2	1
16	58	0	3	150	283	1	0	162	0	1	2	0	2	1
17	50	0	2	120	219	0	1	158	0	1.6	1	0	2	1
18	58	0	2	120	340	0	1	172	0	0	2	0	2	1
19	66	0	3	150	226	0	1	114	0	2.6	0	0	2	1

Independent variables that are potential predictors (e.g., age, cholesterol levels, blood pressure). Dependent variable indicating the presence (1) or absence (0) of heart disease.

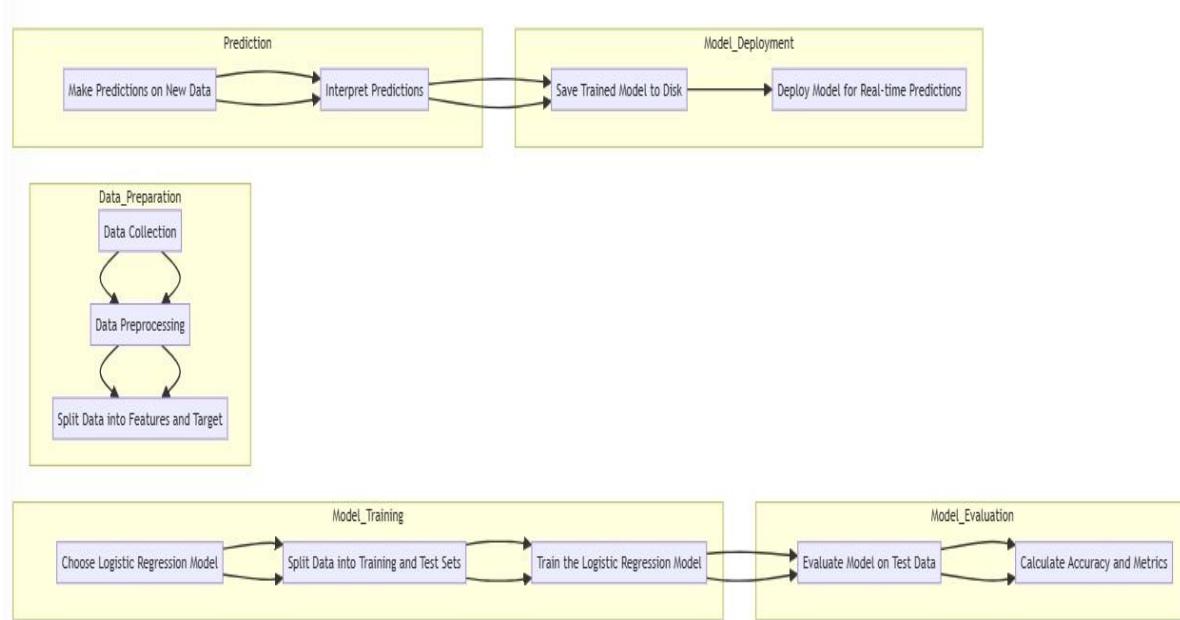
Logistic Regression is a common classification algorithm used for binary classification tasks like heart disease prediction.

Logistic Regression models the probability of the target variable using the sigmoid function:

$$P(Y = 1|X) = \frac{1}{1+e^{-\beta^T X}} \quad \text{where } \beta \text{ are the coefficients learned during training.}$$

Data is divided into training and testing sets to evaluate model performance. Used to measure model performance based on the proportion of correct predictions.

After training, the model can predict heart disease risk for new input data (e.g., a person's vital statistics). Based on the model's prediction probability, individuals can be categorized into high or low-risk groups. Saving the trained model allows for future use without retraining, enabling quick predictions on new data.



The provided Python code snippet demonstrates these concepts by:

Loading the heart disease dataset.

Preprocessing the data (splitting into training and test sets).

Training a logistic regression model.

Evaluating the model's accuracy.

Making predictions on new data.

Saving and loading the trained model for future use.

In summary, heart disease prediction using ML involves a comprehensive understanding of statistical modeling, feature engineering, model training, evaluation, and deployment.

Logistic regression is one of the fundamental algorithms used for this purpose due to its simplicity and interpretability in binary classification tasks. However, more complex algorithms and techniques can also be employed depending on the dataset and problem requirements.

2. Breast Cancer Prediction:

Breast cancer prediction using machine learning (ML) involves leveraging statistical methods and algorithms to analyze medical data and predict the presence or absence of breast cancer based on relevant features. Breast cancer is a significant health concern, and ML can aid in early detection and diagnosis by analyzing patient data and identifying patterns associated with cancerous tumors.

The dataset typically contains various attributes (features) related to breast cancer, such as clump thickness, cell size, cell shape, adhesion, etc., along with a target variable indicating the diagnosis (e.g., benign or malignant).

Logistic Regression is a widely used ML algorithm for binary classification tasks like breast cancer prediction. It models the probability of the target variable (malignant or benign) given the input features using a sigmoid function:

$$P(Y = 1 | X) = \frac{1}{1+e^{-\beta^T X}}$$

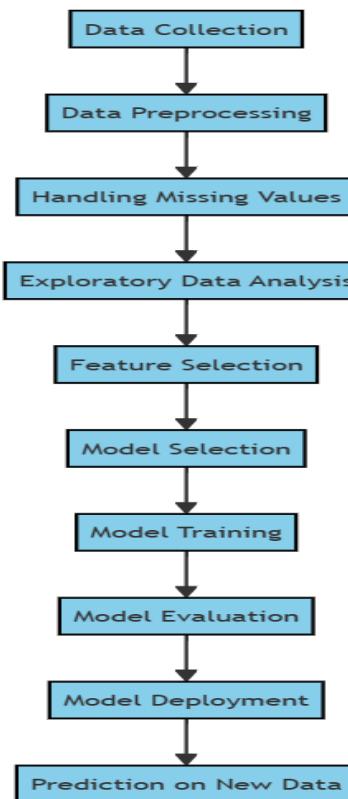
- Y is the target variable (1 for malignant, 0 for benign).
- X is the vector of input features.
- β are the model parameters (coefficients) learned during training.

Split the dataset into training and testing sets. Use the training data to fit the logistic regression model. Assess the model's performance using metrics like accuracy, precision, recall, and confusion matrix. Address missing data by imputation or removal. Normalize or standardize features to ensure uniformity.

Compute descriptive statistics (mean, median, etc.) and visualize data distributions.

Compute descriptive statistics (mean, median, etc.) and visualize data distributions.

Select relevant features based on correlation with the target variable. Use techniques like PCA (Principal Component Analysis) for feature reduction. Experiment with various ML algorithms (e.g., Decision Trees, SVM, KNN) to identify the best-performing model. Utilize cross-validation techniques (e.g., k-fold cross-validation) for robust model evaluation.



This approach showcases a systematic workflow for breast cancer prediction using ML, incorporating essential steps from data preparation and feature engineering to model training, evaluation, and deployment.

3. Diabetes Prediction:

Diabetes disease prediction using machine learning (ML) involves leveraging statistical models and algorithms to analyze medical data and predict the likelihood of a person having diabetes based on certain features.

Diabetes is a chronic disease that affects how your body turns food into energy. ML can be used to predict diabetes risk based on patient characteristics and health indicators.

	A	B	C	D	E	F	G	H	I
1	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
2	6	148	72	35	0	33.6	0.627	50	1
3	1	85	66	29	0	26.6	0.351	31	0
4	8	183	64	0	0	23.3	0.672	32	1
5	1	89	66	23	94	28.1	0.167	21	0
6	0	137	40	35	168	43.1	2.288	33	1
7	5	116	74	0	0	25.6	0.201	30	0
8	3	78	50	32	88	31	0.248	26	1
9	10	115	0	0	0	35.3	0.134	29	0
10	2	197	70	45	543	30.5	0.158	53	1
11	8	125	96	0	0	0	0.232	54	1
12	4	110	92	0	0	37.6	0.191	30	0
13	10	168	74	0	0	38	0.537	34	1
14	10	139	80	0	0	27.1	1.441	57	0
15	1	189	60	23	846	30.1	0.398	59	1
16	5	166	72	19	175	25.8	0.587	51	1
17	7	100	0	0	0	30	0.484	32	1
18	0	118	84	47	230	45.8	0.551	31	1
19	7	107	74	0	0	29.6	0.254	31	1
20	1	103	30	38	83	43.3	0.183	33	0
21	1	115	70	30	96	34.6	0.529	32	1
22	3	126	88	41	235	39.3	0.704	27	0
23	8	99	84	0	0	35.4	0.388	50	0
24	7	196	90	0	0	39.8	0.451	41	1
25	9	119	80	35	0	29	0.263	29	1
26	11	143	94	33	146	36.6	0.254	51	1
27	10	125	70	26	115	31.1	0.205	41	1
28	7	147	76	0	0	39.4	0.257	43	1
29	1	97	66	15	140	23.2	0.487	22	0
30	13	145	82	19	110	22.2	0.245	57	0
31	5	117	92	0	0	34.1	0.337	38	0
32	5	109	75	26	0	36	0.546	60	0

The dataset typically includes patient attributes (e.g., glucose level, blood pressure, BMI) and a target variable indicating diabetes status (0 for non-diabetic, 1 for diabetic).

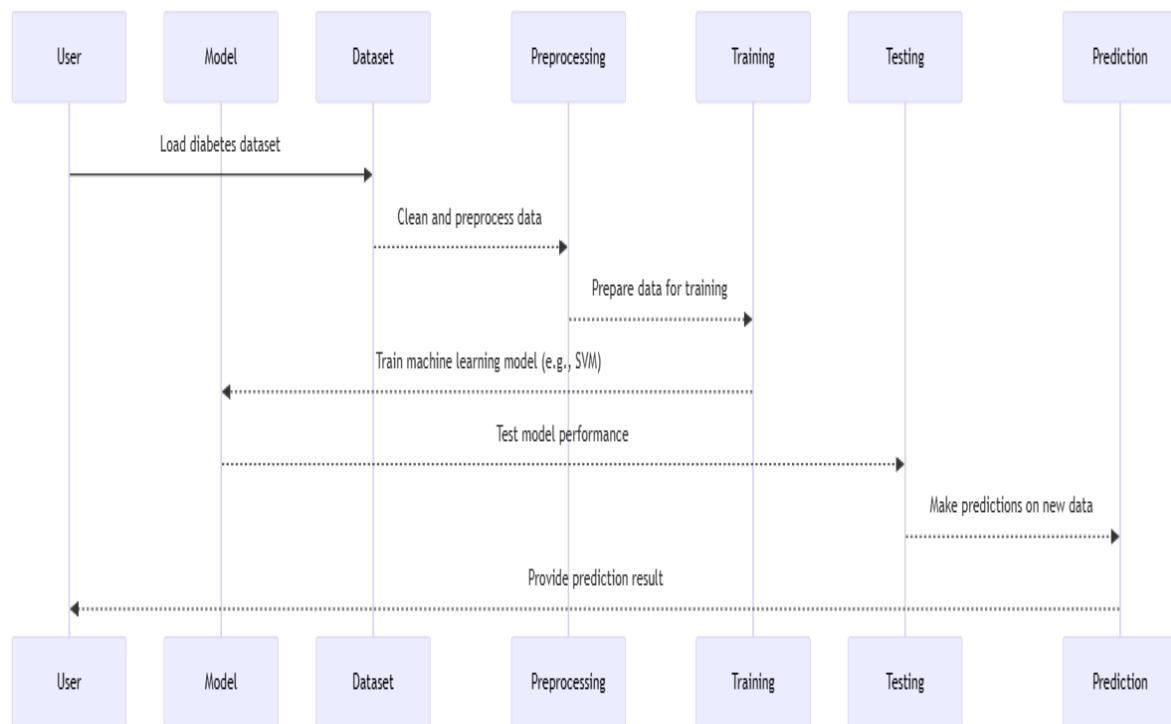
Logistic Regression is a suitable ML algorithm for binary classification tasks like diabetes prediction. It models the probability of a binary outcome (diabetic or non-diabetic) using a logistic function:

$$P(Y = 1 | X) = \frac{1}{1 + e^{-\beta^T X}}$$

- Y is the binary target variable (1 for diabetic, 0 for non-diabetic).
- X is the vector of input features.
- β are the model parameters (coefficients) learned during training.

Addressing missing data through imputation or removal. Normalizing or standardizing features to ensure uniformity.

Splitting the dataset into training and testing sets. Fitting the logistic regression model using the training data. Assessing the model's accuracy and performance using evaluation metrics (e.g., accuracy score, confusion matrix). Saving the trained model for future use and deployment in real-world applications.



Using the deployed model to make predictions on new data (e.g., patient information) to determine diabetes risk.

This approach illustrates a systematic workflow for diabetes prediction using ML techniques, from data preparation and model training to deployment and prediction on new instances.

4. Parkinsons Disease Prediction:

Parkinson's Disease prediction using machine learning (ML) involves utilizing statistical models and algorithms to analyze medical data and predict the likelihood of a person having Parkinson's Disease based on relevant features.

Parkinson's Disease is a progressive nervous system disorder that affects movement. ML can be applied to predict the presence or absence of Parkinson's Disease based on patient characteristics and clinical indicators.

	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	radius_me	texture_me	perimeter	area_me	smoothne	compactn	concavity	concave p	symmetry	fractal_d	radius_se	texture_se	perimeter	area_se	smoothne_se	compactn_se	concavity_se	concave p_se	symmetry_se	fractal_d_se	radius_wc	texture_wc	perimeter
2	17.99	10.38	122.8	1001	0.1184	0.2776	0.3001	0.1471	0.2419	0.0781	1.095	0.9053	8.589	153.4	0.006399	0.04904	0.05373	0.01587	0.03003	0.006193	25.38	17.33	1
3	20.57	17.77	132.9	1326	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	0.5435	0.7339	3.398	74.08	0.005225	0.01308	0.0186	0.0134	0.01389	0.003532	24.99	23.41	1
4	19.69	21.25	130	1203	0.1096	0.1599	0.1974	0.1279	0.2069	0.05999	0.7456	0.7869	4.585	94.03	0.00615	0.04006	0.03832	0.02058	0.0225	0.004571	23.57	25.53	1
5	11.42	20.38	77.58	386.1	0.1425	0.2839	0.2414	0.1052	0.2597	0.09744	0.4956	1.156	3.445	27.23	0.00911	0.07458	0.05661	0.01867	0.05963	0.009208	14.91	26.5	9
6	20.29	14.34	135.1	1297	0.1003	0.1328	0.198	0.1043	0.1809	0.05883	0.7572	0.7813	5.438	94.44	0.01149	0.02461	0.05688	0.01885	0.01756	0.005115	22.54	16.67	1
7	12.45	15.7	82.57	477.1	0.1278	0.17	0.1578	0.08089	0.2087	0.07613	0.3345	0.6902	2.217	27.19	0.00751	0.03345	0.03672	0.01137	0.02165	0.005082	15.47	23.75	1
8	18.25	19.98	119.6	1040	0.09463	0.109	0.1127	0.074	0.1794	0.05742	0.4467	0.7732	3.18	53.91	0.004314	0.01382	0.02254	0.01039	0.01369	0.002179	22.88	27.66	1
9	13.71	20.83	90.2	577.9	0.1189	0.1645	0.09366	0.05985	0.2198	0.07451	0.5835	1.377	3.856	50.96	0.008805	0.03029	0.02488	0.01448	0.005412	17.06	28.14	1	
10	13	21.82	87.5	519.8	0.1273	0.1932	0.1859	0.09353	0.235	0.07389	0.3063	1.002	2.406	24.32	0.005731	0.03502	0.035532	0.01226	0.02143	0.003749	15.49	30.73	1
11	12.46	24.04	83.97	475.9	0.1186	0.2396	0.2273	0.08543	0.203	0.08243	0.2976	1.599	2.039	23.94	0.007149	0.07217	0.07743	0.01432	0.01789	0.01008	15.09	40.68	9
12	16.02	23.24	102.7	797.8	0.08206	0.06660	0.03299	0.03323	0.1528	0.05697	0.3795	1.187	2.466	40.51	0.004029	0.009269	0.01101	0.007591	0.0148	0.003042	19.19	33.88	1
13	15.78	17.89	103.6	781	0.0971	0.1292	0.09954	0.06606	0.1842	0.06082	0.5058	0.9849	3.564	34.16	0.005771	0.04061	0.02791	0.01282	0.02008	0.004144	20.42	27.28	1
14	19.17	24.8	132.4	1123	0.0974	0.2458	0.2065	0.1118	0.2397	0.078	0.9555	3.568	11.07	116.2	0.003139	0.08297	0.0889	0.0409	0.04484	0.01284	20.96	29.94	1
15	15.85	23.95	103.7	782.7	0.08401	0.1002	0.09938	0.05364	0.1847	0.05338	0.4033	1.078	2.903	36.58	0.009769	0.03126	0.05051	0.01992	0.02981	0.003002	16.84	27.66	1
16	13.73	22.61	93.6	578.3	0.1131	0.2293	0.2128	0.08025	0.2069	0.07682	0.2121	1.169	2.061	19.21	0.006429	0.05936	0.054501	0.01628	0.01961	0.008093	15.03	32.01	1
17	14.54	27.54	96.73	658.8	0.1139	0.1595	0.1639	0.07364	0.2303	0.07077	0.37	1.033	2.879	32.55	0.005607	0.0424	0.04741	0.0109	0.01857	0.005466	17.46	37.13	1
18	14.68	20.13	94.74	684.5	0.09867	0.072	0.07395	0.05259	0.1586	0.05922	0.4727	1.24	3.195	45.4	0.005718	0.01162	0.01998	0.01109	0.0141	0.002085	19.07	30.88	1
19	16.13	20.68	108.1	798.8	0.117	0.2022	0.1722	0.1028	0.2164	0.07356	0.5692	1.073	3.854	54.18	0.007026	0.00968	0.03188	0.01297	0.01688	0.004142	20.96	31.48	1
20	19.81	22.15	130	1260	0.09831	0.1027	0.09498	0.09498	0.1582	0.05395	0.7582	1.017	5.865	112.4	0.006494	0.01893	0.03391	0.01521	0.01356	0.001997	27.32	30.88	1
21	13.54	14.36	87.46	566.3	0.09779	0.08128	0.06664	0.04781	0.1885	0.05766	0.2699	0.7886	2.058	23.56	0.008462	0.0146	0.02387	0.01315	0.0198	0.0023	15.11	19.26	1
22	13.08	15.71	85.63	520	0.1075	0.127	0.04568	0.0311	0.1967	0.08111	0.1852	0.7477	1.383	14.67	0.004097	0.01898	0.01698	0.00649	0.01678	0.002425	14.5	20.49	9
23	9.504	12.44	60.34	273.9	0.1024	0.06492	0.02956	0.02076	0.1815	0.06905	0.2773	0.9768	1.909	15.7	0.009606	0.01432	0.01985	0.01421	0.02027	0.002968	10.23	15.66	6
24	15.34	14.26	102.5	704.4	0.1073	0.2135	0.2077	0.09756	0.2521	0.07032	0.4388	0.7096	3.384	44.91	0.007689	0.05328	0.06446	0.02252	0.03672	0.004394	18.07	19.08	1
25	21.16	23.04	137.2	1404	0.09428	0.1022	0.1097	0.08632	0.1769	0.05278	0.6917	1.127	4.303	93.99	0.004728	0.01259	0.01715	0.01038	0.01083	0.001987	29.17	35.59	1
26	16.65	21.38	110	904.6	0.1121	0.1457	0.1525	0.0917	0.1995	0.0633	0.8068	0.9017	5.455	102.6	0.006048	0.01882	0.02741	0.0113	0.01468	0.002801	26.46	31.56	1
27	17.14	16.4	116	912.7	0.1186	0.2276	0.2229	0.1401	0.304	0.07413	1.046	0.976	7.276	111.4	0.008029	0.03799	0.03732	0.02397	0.02308	0.007444	22.25	21.4	1
28	14.58	21.53	97.41	644.0	0.1054	0.1868	0.1425	0.08783	0.2522	0.06924	0.2545	0.9832	2.11	21.05	0.004452	0.03055	0.02681	0.01352	0.01454	0.003711	17.62	33.21	1
29	18.61	20.25	122.1	1094	0.0944	0.1066	0.149	0.07731	0.1697	0.05699	0.8529	1.849	5.632	93.54	0.01075	0.02722	0.05081	0.01911	0.02293	0.004217	21.31	27.26	1
30	15.3	25.27	102.4	732.4	0.1082	0.1697	0.1683	0.08751	0.1926	0.06754	0.439	1.012	3.498	43.5	0.005233	0.03057	0.03576	0.01083	0.01768	0.002967	20.27	36.71	1
31	17.57	15.05	115	955.1	0.09847	0.1157	0.09875	0.07953	0.1739	0.06149	0.6003	0.8225	4.655	61.1	0.005627	0.03033	0.03407	0.01354	0.01925	0.003742	20.01	19.52	1
32	18.63	25.11	124.8	1088	0.1064	0.1887	0.2319	0.1244	0.2183	0.06197	0.8307	1.466	5.574	105	0.006248	0.03374	0.05196	0.01158	0.02007	0.00456	23.15	34.01	1

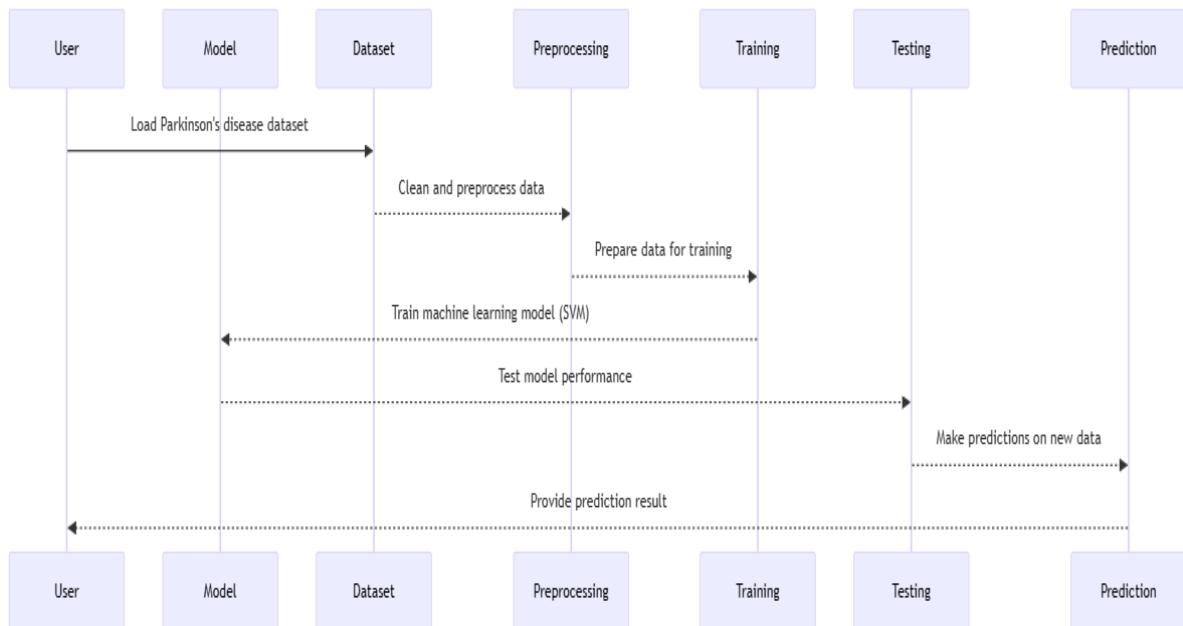
The dataset typically includes features (e.g., voice recordings, clinical measurements) and a target variable indicating the disease status (0 for non-Parkinson's, 1 for Parkinson's).

Support Vector Machine (SVM) is a powerful ML algorithm used for binary classification tasks like Parkinson's Disease prediction. SVM aims to find the optimal hyperplane that best separates the classes by maximizing the margin between them.

Addressing missing data through imputation or removal. Normalizing or standardizing features to ensure uniformity.

Normalizing or standardizing features to ensure uniformity.

Splitting the dataset into training and testing sets. Fitting the SVM model using the training data. Assessing the model's accuracy and performance using evaluation metrics (e.g., accuracy score, confusion matrix).



Using the trained model to make predictions on new data (e.g., clinical measurements) and interpret the results.

- The dataset is loaded and preprocessed.
- Features and target data are separated.
- The SVM model

Data Collection & Analysis (A): Collect and analyze Parkinson's disease dataset.

Data Preprocessing (B): Clean, preprocess, and prepare the dataset for modeling.

Separate Features & Target Data (C): Divide the dataset into features (input variables) and target data (status).

Split Data into Training & Test Sets (D): Split the dataset into training and test sets for model training and evaluation.

Model Training (SVM) (E): Train a Support Vector Machine (SVM) model using the training dataset.

Model Evaluation (F): Evaluate the trained model's performance using the test dataset.

Prediction (G): Make predictions using the trained model on new data.

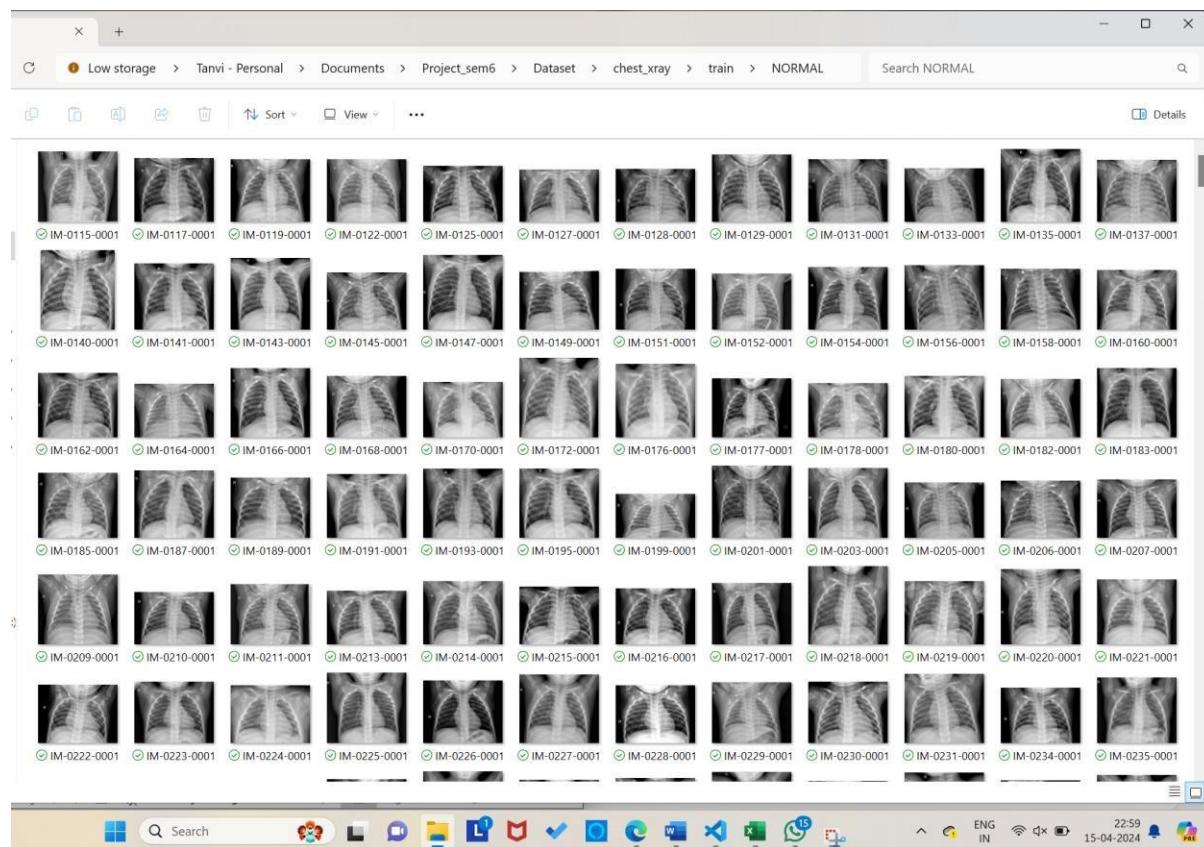
5. Pneumonia Disease Prediction:

Predicting pneumonia disease using deep learning involves leveraging neural network architectures to analyze medical imaging data (e.g., chest X-rays) and classify whether a patient has pneumonia or not.

Pneumonia is a common and potentially serious lung infection caused by bacteria, viruses, or fungi. Deep learning can be used to analyze chest X-ray images and automatically detect signs of pneumonia.

Convolutional Neural Networks (CNNs) are widely used in image analysis tasks, including medical image classification. CNNs are designed to automatically learn hierarchical patterns and features from image data, making them effective for pneumonia detection.

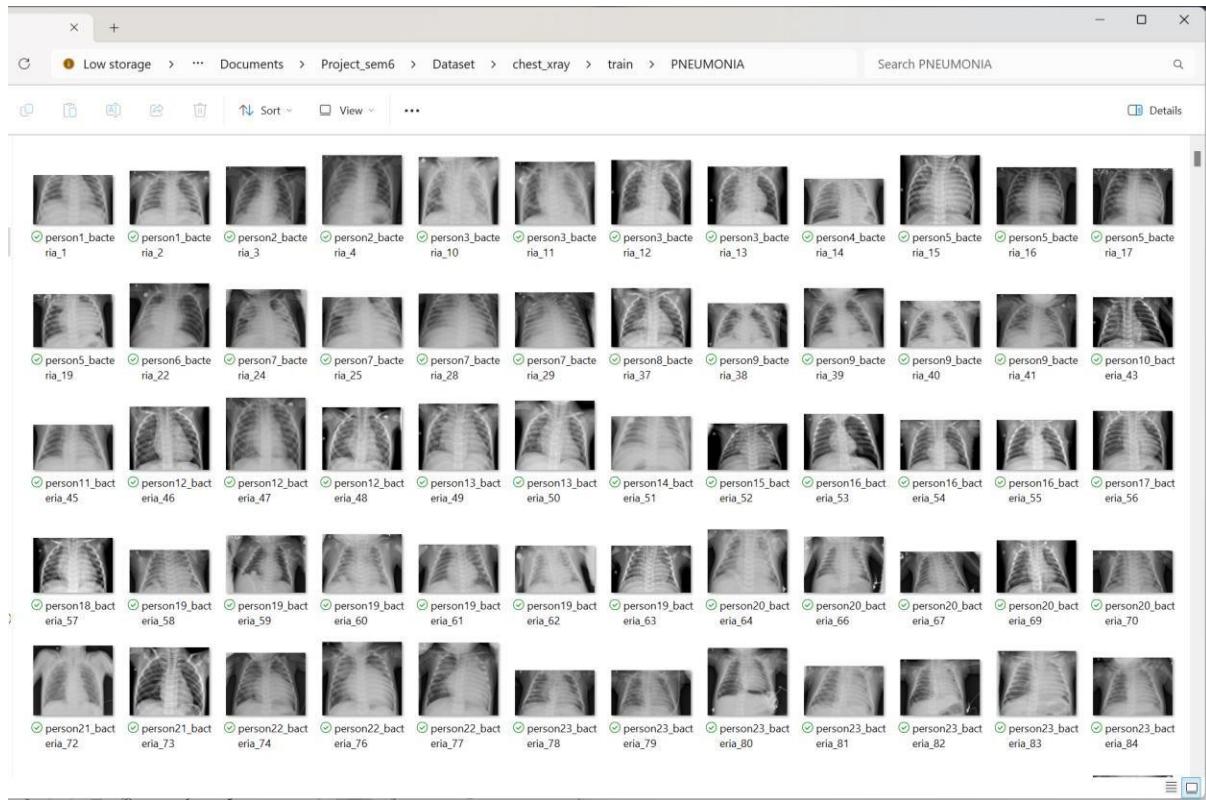
The X-Ray for Normal Patient is :



The dataset typically consists of chest X-ray images labeled with pneumonia status (positive or negative). Each image serves as an input to the deep learning model.

Standardize the size of input images for consistency. Scale pixel values to a common range (e.g., 0 to 1) for efficient model training.

The X-Ray for Pneumonia Patient is :

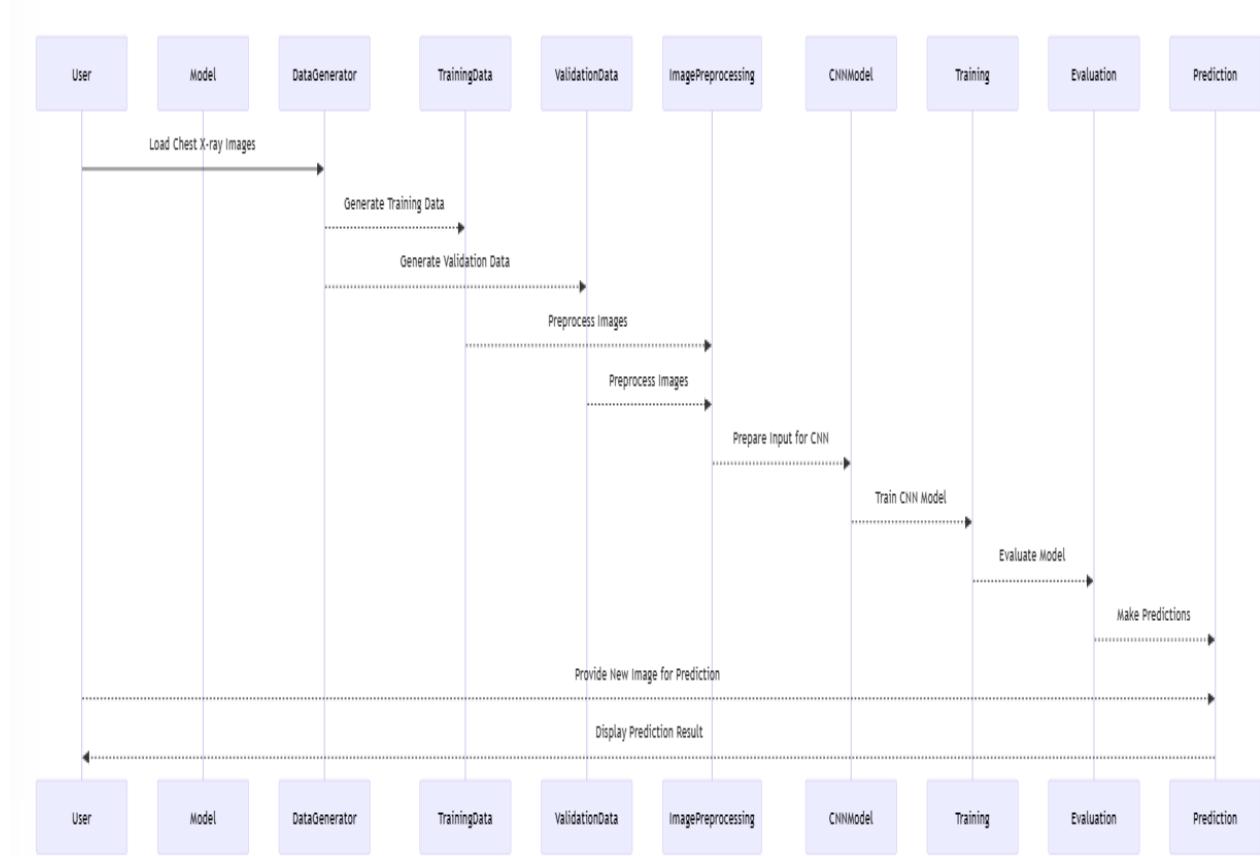


Construct a CNN architecture consisting of convolutional layers, pooling layers, and fully connected layers. Use activation functions like ReLU (Rectified Linear Unit) to introduce non-linearity. Utilize a softmax output layer for multi-class classification (pneumonia or not pneumonia).

Split the dataset into training and validation sets. Define a suitable loss function (e.g., cross-entropy) for training the model. Choose an optimizer (e.g., Adam, SGD) to minimize the loss during training. Evaluate model performance using metrics such as accuracy, precision, recall, and F1-score.

- Chest X-ray images are loaded and preprocessed using an `ImageDataGenerator`.
- A CNN model is constructed using `Sequential` model from TensorFlow/Keras.
- The model is trained on the training dataset and evaluated using the validation dataset.
- Model performance metrics (loss, accuracy) are plotted using `matplotlib`.
- New chest X-ray images can be predicted using the trained model to detect pneumonia.

Use the trained deep learning model to predict pneumonia status on new chest X-ray images. Interpret the model's predictions and provide diagnostic insights.



This implementation demonstrates the application of deep learning for pneumonia disease prediction using CNNs in Python. Customize the model architecture, training parameters, and evaluation metrics based on specific dataset characteristics and requirements for accurate pneumonia detection.

Initiates the process by loading chest X-ray images and requesting predictions for new images. Generates training and validation data from loaded chest X-ray images. Represents the datasets used for training and validation of the CNN model. Handles image preprocessing tasks such as resizing and normalization before inputting into the CNN. Represents the convolutional neural network model used for pneumonia prediction. Trains the CNN model using the preprocessed training data. Evaluates the trained model's performance using the validation dataset. Utilizes the trained CNN model to make predictions on new chest X-ray images provided by the user.

Chapter 4 : Coding

Interface (main)

the MyMedCare interface, designed to optimize patient care and streamline healthcare management. Our platform integrates user-friendly features tailored for healthcare providers, administrators, and patients alike, ensuring a seamless experience across all interactions.

Dashboard

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>MyMedCare</title>
    <link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
        <script
            src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <script
        src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js"
    "></script>
    <script
        src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
    <script
        src="https://use.fontawesome.com/releases/v5.0.8/js/all.js"></script>
    <link rel="stylesheet" type="text/css" href="static/css/aboutus.css">
    <link rel="canonical"
        href="https://getbootstrap.com/docs/4.5/examples/carousel/">
```

declaration specifies that the document type is HTML5, ensuring the browser renders the document correctly.

The `<html>` element is the root element of the HTML document, and the `lang="en"` attribute specifies that the document is in English.

The `<head>` section contains metadata and external resources used by the HTML document. This includes character encoding, viewport settings, title, external CSS stylesheets, and JavaScript libraries.

- `<meta charset="utf-8">`: Specifies the character encoding of the document as UTF-8.
- `<title>MyMedCare</title>`: Sets the title of the HTML document.
- `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">`: Links to the Bootstrap CSS framework to style the HTML content.
- `<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>`: Loads the jQuery library.

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <title>MyMedCare</title>
6      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
7      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
8      <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"></script>
9      <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
10     <script src="https://use.fontawesome.com/releases/v5.0.8/js/all.js"></script>
11     <link rel="stylesheet" type="text/css" href="static/css/aboutus.css">
12     <link rel="canonical" href="https://getbootstrap.com/docs/4.5/examples/carousel/">
13

```

- `<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"></script>`: Loads the Popper.js library (required for Bootstrap tooltips, popovers, etc.).
- `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>`: Loads the Bootstrap JavaScript library.
- `<script src="https://use.fontawesome.com/releases/v5.0.8/js/all.js"></script>`: Loads Font Awesome icons.
- `<link rel="stylesheet" type="text/css" href="static/css/aboutus.css">`: Links to a custom CSS file (aboutus.css) to further style the HTML content.

- `<link rel="canonical" href="https://getbootstrap.com/docs/4.5/examples/carousel /">`: Specifies the canonical URL for search engines, indicating the preferred version of the webpage.

```

<style>
    body {
        padding-top: 3.5rem;
    }

    nav {
        font-family: sans-serif;
    }

    .navbar-brand {
        font-size: 20px;
        font-family: Georgia, 'Times New Roman', Times, serif;
    }
</style>

```

Styling for `body` Element:

`padding-top: 3.5rem;`: This style adds a top padding of 3.5rem (or 56 pixels) to the `<body>` element. This can be useful for creating space at the top of the page, especially when using a fixed navigation bar.

`font-family: sans-serif;`: This style sets the font family for all `<nav>` elements to a generic sans-serif font. This will apply to any `<nav>` element within the document.

`.navbar-brand`: This targets elements with the class `navbar-brand`, which is commonly used for branding logos or text within Bootstrap navigation bars.

`font-size: 20px;`: This sets the font size of elements with the `.navbar-brand` class to 20px.

`font-family: Georgia, 'Times New Roman', Times, serif;`: This specifies a font stack for the `.navbar-brand` elements. It will attempt to use the Georgia font first, followed by Times New Roman, Times, and finally a generic serif font if none of the specified fonts are available.

- The padding-top applied to the <body> element creates space at the top of the page, accommodating the fixed navigation bar (if present).
- The font-family style applied to <nav> elements ensure a consistent sans-serif font for navigation components.
- The styles for .navbar-brand ensure that navigation bar branding elements have a specific font size and font family, providing a customized look for the brand logo or text.

```

<body>
  <form id="form1">
    <header>
      <!--Navigation Bar-->
      <nav class="navbar navbar-expand-md navbar-dark fixed-top bg-dark">
        <a class="navbar-brand" href="#" style="color: rgb(9, 240, 9);"><b>MyMedCare</b></a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarCollapse" aria-controls="navbarCollapse" aria-expanded="false" aria-label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarCollapse">
          <ul class="navbar-nav ml-auto">

            <li class="nav-item">
              <a class="nav-link" href="C:\Users\SLIM5\OneDrive\Documents\Project_sem6\HTML_Files\diseaseindex.html" style="color: rgb(224, 227, 234);">Disease Index</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="http://192.168.0.103:8501" style="color: rgb(224, 227, 234);">Breast Cancer</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="http://192.168.0.103:8502" style="color: rgb(224, 227, 234);">Diabetes</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="http://192.168.0.103:8503" style="color: rgb(224, 227, 234);">Heart Disease</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="http://192.168.0.103:8504" style="color: rgb(224, 227, 234);">Parkinsons Disease</a>
            </li>
            <li class="nav-item">

```

```

        <a class="nav-link" href="http://192.168.0.103:8505"
style="color: rgb(224, 227, 234);">Pneumonia Disease</a>
    </li>
</ul>
</div>
</nav>

<main role="main">
```

- **<body>**: This is the main content area of the HTML document where visible content is placed.
- **<form id="form1">**: The `<form>` element defines an HTML form with the ID attribute set to "form1". Forms are used to collect user input.
- **<header>**: The `<header>` element represents introductory content, typically containing navigation links, logos, and headings.
- **<nav>**: The `<nav>` element defines a navigation bar (navbar) using Bootstrap classes. It includes a brand link (``) and a collapsible menu (`<div class="collapse navbar-collapse">`) with navigation items (`<ul class="navbar-nav">`).
- ****: This is a link styled as the brand logo or name of the website (MyMedCare), positioned within the navigation bar.
- **<button class="navbar-toggler">**: This button toggles the visibility of the collapsible menu on smaller screens (e.g., mobile devices).
- **<div class="collapse navbar-collapse">**: This `<div>` contains the collapsible menu items (defined within `<ul class="navbar-nav">`) that are displayed when the toggle button is clicked.
- **<ul class="navbar-nav">**: This `` element represents an unordered list of navigation items (`` elements with `<a>` tags) within the navbar.
- **<main role="main">**: The `<main>` element specifies the main content of the webpage. It has the role attribute set to "main" for accessibility purposes.
- Navigation Links:
- The `<a>` elements (``) inside the navbar (`<ul class="navbar-nav">`) represent navigation links to different sections or pages of the website. Each `<a>` tag has an `href` attribute that specifies the destination URL.

- Example: `Breast Cancer`
- This link (Breast Cancer) navigates to http://192.168.0.103:8501 when clicked.
- `The style="color: rgb(224, 227, 234);"` attribute sets the text color to a specific shade of gray (rgb(224, 227, 234)).

```

<div class="jumbotron">
  <div class="container">
    <h1 class="display-4 text-center" style="color: white; background-color: darkgreen; font-family: 'Times New Roman', Times, serif; font-weight: bold;"><b>MyMedCare</b></h1>

    <p class="display-4 text-center" style="font-size: 1.5rem; color: rgb(2, 49, 84);">"Welcome to our innovative platform dedicated to disease prediction and personalized healthcare recommendations. At MyMedCare, we utilize cutting-edge technology to forecast potential health risks and provide tailored suggestions for preventive measures.

    Our advanced algorithms analyze various factors to predict the likelihood of specific diseases, empowering users to take proactive steps towards their well-being. Additionally, our platform offers curated recommendations for doctors and hospitals based on individual needs, ensuring access to quality healthcare services. Join us in taking charge of your health journey today."</p>

  </div>
</div>

```

- `<div class="jumbotron">`: This is a Bootstrap component used to showcase key content, often placed at the top of a page or section. It has predefined styling to create a large, prominent container.
- `<div class="container">`: This `<div>` element is nested within the jumbotron and uses Bootstrap's `.container` class to center the content and limit its width based on the device's screen size.
- `<h1 class="display-4 text-center" style="color: white; background-color: darkgreen; font-family: 'Times New`

Roman', Times, serif; font-weight: bold;">" + " + "

MyMedCare " + " + "</h1>:

- <h1>: This is a heading element (<h1>) with the class display-4, which applies a large, bold style to the text.
- text-center: This Bootstrap class aligns the text to the center within its container.
- style="color: white; background-color: darkgreen; font-family: 'Times New Roman', Times, serif; font-weight: bold;":
- color: white;: Sets the text color to white.
- background-color: darkgreen;: Sets the background color to dark green.
- font-family: 'Times New Roman', Times, serif;: Specifies the font family for the text.
- font-weight: bold;: Makes the text bold.
- <p class="display-4 text-center" style="font-size: 1.5rem; color: rgb(2, 49, 84);">...</p>:
- <p>: This is a paragraph element (<p>) with the class display-4, which applies a large, bold style to the text.
- text-center: Aligns the text to the center within its container.
- style="font-size: 1.5rem; color: rgb(2, 49, 84);":
- font-size: 1.5rem;: Sets the font size to 1.5 times the default size.
- color: rgb(2, 49, 84);: Sets the text color using RGB values.

```
63 |     |         |<main role="main">
64 |     |         |
65 |     |         |<div class="jumbotron">
66 |     |         |<div class="container">
67 |     |         |<h1 class="display-4 text-center" style="color: white; background-color: darkgreen; font-family: 'Times New Roman', Times, serif; font-weight: bold;">MyMedCare " + " + "</h1>
68 |     |         |<p class="display-4 text-center" style="font-size: 1.5rem; color: rgb(2, 49, 84);">"Welcome to our innovative platform we utilize cutting-edge technology to forecast potential health risks and provide tailored suggestions for preventive medicine. Our advanced algorithms analyze various factors to predict the likelihood of specific diseases, empowering users to take proactive steps towards their well-being. Additionally, our platform offers curated recommendations for doctors and hospitals based on individual needs, ensuring access to quality healthcare services. Join us in taking charge of your health journey today."</p>
69 |     |         |
70 |     |         |</div>
71 |     |         |
72 |     |         |</div>
73 |     |         |
74 |     |         |
75 |     |         |
76 |     |         |
77 |     |         |
78 |     |         |</div>
```

- The content within the <div class="jumbotron"> consists of a heading (<h1>) and a paragraph (<p>):
- The heading (<h1>) displays the text " " + " + MyMedCare " + " + " with customized styling (white text on dark green background, bold font).

- The paragraph (<p>) provides a welcome message and overview of the MyMedCare platform, describing its features and benefits in disease prediction and personalized healthcare recommendations.

```
<div>
  <p>
    <img src = "C:\Users\SLIM5\OneDrive\Documents\Project_sem6\New
folder\fullfledge.jpg" alt = "Image" class="img-fluid" >
  </p>
</div>
```

Use Relative Paths: Instead of using an absolute file path

(`src="C:\Users\SLIM5\OneDrive\Documents\Project_sem6\New
folder\fullfledge.jpg"`) , consider using a relative path if the image file is located within the same directory or a subdirectory of your HTML document.

This assumes that the fullfledge.jpg image is located within an images directory relative to your HTML document.

```
<div>
  <p>
    <img src = "C:\Users\SLIM5\OneDrive\Documents\Project_sem6\New folder\fullfledge.jpg" alt = "Image" class="img-fluid" >
  </p>
</div>
<div>
```

For web deployment, it's recommended to host images online (using services like Imgur, AWS S3, or your website's server) and use URLs

(`src="https://example.com/path/to/image.jpg"`) instead of local file paths.

```
div>
  <h4><p><marquee width="100%" behavior="scroll">
    <font face="Bahnschrift Condensed" size="10"
color="darkgreen">"Predict. Protect. Prevent"</font></marquee></p></h4>
  <p><marquee width="100%" behavior="scroll" direction="right">
    <font face="Bahnschrift Condensed" size="6"
color="darkblue">"Prediction and Recommendation for your Life"</font>
  </marquee></p>
</div>
```

- **<div>**: This is a <div> (division) element, used as a container to group and style the enclosed content.
- **<h4>**: This is a heading level 4 (<h4>) element, which represents a subheading.
- **<p>**: This is a paragraph (<p>) element, used here as a wrapper within the <h4> block.
- **<marquee>**: This is a non-standard HTML element used to create scrolling text or images.

```
<div>
  <h4><p><marquee width="100%" behavior="scroll">
    <font face="Bahnschrift Condensed" size="10" color="darkgreen">"Predict. Protect. Prevent"</font></marquee></p></h4>
  <p><marquee width="100%" behavior="scroll" direction="right">
    <font face="Bahnschrift Condensed" size="6" color="darkblue">"Prediction and Recommendation for your Life"</font>
  </marquee></p>
</div>
```

- **width="100%"** : Sets the width of the <marquee> to 100% of its container.
- **behavior="scroll"**: Specifies the scrolling behavior of the text within the <marquee>.
- ****: This is a deprecated HTML element used to specify font-related attributes such as face (font family), size, and color. It's not recommended for use in modern HTML and CSS.
- **face="Bahnschrift Condensed"**: Sets the font family to "Bahnschrift Condensed".
- **size="10"** : Sets the font size to 10 (deprecated; use CSS for font sizing).
- **color="darkgreen"** : Sets the text color to dark green.

Instead of using **** for styling, use CSS to apply font styles (font family, size, color) to text elements (**<h4>**, **<p>**, etc.)

Instead of **<marquee>**, use CSS animations or JavaScript libraries (like jQuery) for more modern and controlled scrolling effects.

Use HTML elements (**<h4>**, **<p>**) appropriately based on the content's meaning and structure.

Ensure that scrolling text and other dynamic content adapt well to different screen sizes and devices.

```
<h><div class="container">
    <div class="row">
        <div class="col-md-6">
            
        </div>
        <div class="col-md-6">
            <div class="about-us">
                <h2>About Us</h2>
                <p>Welcome to MyMedCare, your trusted partner in
healthcare.
```

Our mission is to revolutionize the way you approach wellness by providing personalized, data-driven solutions.

With a team of dedicated experts, we harness the power of technology to empower individuals to make informed decisions about their health.

Through continuous innovation and research, we strive to stay at the forefront of preventive healthcare, ensuring you have access to the latest advancements in the field.

At MyMedCare, we are committed to your well-being and are here to support you every step of your health journey.</p>

```
    <table>
        <tr>
            <td><h5 style="color: rgb(14, 84,
21);"><b>Predict</b></h5></td>
        </tr>
        <tr>
            <td>Over here a user can add its vitals and it
will forecast the disease with its Potential Health Conditions and whether the
patient is likely to be diagnosed with that disease. Diseases such as Breast
Cancer, Pneumonia, Heart Disease, Diabetes, and Parkinson's Disease can be
predicted here.</td>
        </tr>
        <tr>
            <td><h5 style="color: rgb(14, 84,
21);"><b>Prevent</b></h5></td>
        </tr>
```

```
51
```

```

        <tr>
            <td>Here it will recommend the doctors
according to the disease predicted. It will ask you whether you want to
consult a Specialist or a Hospital with your Vital reports.</td>
        </tr>
        <tr>
            <td><h5 style="color: rgb(14, 84,
21);"><b>Protect</b></h5></td>
        </tr>
        <tr>
            <td>List common symptoms associated with each
disease to help users identify potential health issues. Explain the causes or
risk factors contributing to each disease. </td>
        </tr>
    </table>
</div>
</div>
</div>
</div>
</main>

</form>
</body>
</html>

```

<div class="container">: This `<div>` with the class `container` is a Bootstrap container element, which provides a responsive fixed-width container for content.

<div class="row">: This `<div>` with the class `row` is a Bootstrap row element, used to contain and align columns (`<div class="col-md-6">`) within a grid layout.

<div class="col-md-6">: This `<div>` with the class `col-md-6` is a Bootstrap column element that spans 6 grid columns on medium-sized screens and larger.

****: This `` element embeds an image (`logo1.jpg`) sourced from a local file path

(`C:/Users/SLIM5/OneDrive/Documents/Project_sem6/New folder/logo1.jpg`). The `alt` attribute provides alternative text for accessibility, and the `img-fluid` class makes the image responsive.

<div class="col-md-6">: Another Bootstrap column element that spans 6 grid columns.

<div class="about-us">: A `<div>` with the class `about-us`, likely used to style and contain content related to the "About Us" section.

<h2>About Us</h2>: A heading (**<h2>**) element displaying the title "About Us".

<p>: A paragraph element (**<p>**) containing descriptive text about MyMedCare's mission and services.

<table>: A table element used to organize and present information in tabular format.

Various **<tr>** (table row) and **<td>** (table data/cell) elements containing content related to specific topics like "Predict", "Prevent", and "Protect".

<h5 style="color: #rgb(14, 84, 21);>Predict</h5>: Heading (**<h5>**) styled with specific color and font properties.

<td>...<td>: Table cells containing detailed descriptions of each topic (Predict, Prevent, Protect).

```
91 |     </div>
92 |     </div>
93 |     <div class="container">
94 |         <div class="row">
95 |             <div class="col-md-6">
96 |                 
97 |             </div>
98 |             <div class="col-md-6">
99 |                 <div class="about-us">
100 |                     <h2>About Us</h2>
101 |                     <p>Welcome to MyMedCare, your trusted partner in healthcare.  
Our mission is to revolutionize the way you approach wellness by providing personalized, data-driven sol  
With a team of dedicated experts, we harness the power of technology to empower individuals to make inf  
Through continuous innovation and research, we strive to stay at the forefront of preventive healthcare  
At MyMedCare, we are committed to your well-being and are here to support you every step of your health
102 |                     <table>
103 |                         <tr>
104 |                             <td><h5 style="color: #rgb(14, 84, 21);><b>Predict</b></h5></td>
105 |                         </tr>
106 |                         <tr>
107 |                             <td>Over here a user can add its vitals and it will forecast the disease with its Potential Heal
108 |                         </td>
109 |                         <tr>
110 |                             <td><h5 style="color: #rgb(14, 84, 21);><b>Prevent</b></h5></td>
111 |                         </tr>
112 |                         <tr>
113 |                             <td>Here it will recommend the doctors according to the disease predicted. It will ask you wheth
114 |                         </td>
115 |                         <tr>
116 |                             <td><h5 style="color: #rgb(14, 84, 21);><b>Protect</b></h5></td>
117 |                         </tr>
118 |                         <tr>
119 |                             <td>List common symptoms associated with each disease to help users identify potential health is
120 |                         </td>
121 |                         </tr>
122 |                         <tr>
123 |                             <td>List common symptoms associated with each disease to help users identify potential health is
124 |                         </td>
125 |                         </tr>
126 |                     </table>
127 |                 </div>
128 |             </div>
129 |         </div>
130 |     </div>
131 | 
```

Instead of using absolute file paths (C:/Users/...), consider hosting images online or using relative paths (..../New folder/logo1.jpg) relative to your HTML file. Ensure the use of appropriate HTML elements (**<h2>**, **<p>**, **<table>**) to convey the structure and meaning of your content. Test the responsiveness of your layout across different screen sizes to ensure optimal viewing experience on various devices. Consider separating content and styling by using external CSS stylesheets or internal **<style>** blocks to improve code organization and maintainability.

Overall Input:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>MyMedCare</title>
    <link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
        <script
            src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
        <script
            src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js"
        ></script>
        <script
            src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
        <script
            src="https://use.fontawesome.com/releases/v5.0.8/js/all.js"></script>
        <link rel="stylesheet" type="text/css" href="static/css/aboutus.css">
        <link rel="canonical"
            href="https://getbootstrap.com/docs/4.5/examples/carousel/">

        <style>
            body {
                padding-top: 3.5rem;
            }

        nav {
            font-family: sans-serif;
        }

        .navbar-brand {
            font-size: 20px;
            font-family: Georgia, 'Times New Roman', Times, serif;
        }
        </style>

    </head>
    <body>
        <form id="form1">
            <header>
                <!--Navigation Bar-->
                <nav class="navbar navbar-expand-md navbar-dark fixed-top bg-dark">
                    <a class="navbar-brand" href="#" style="color: #007bff; font-size: 1.2em; margin-right: 10px;">MyMedCare</a>
```

```

        <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#navbarCollapse" aria-controls="navbarCollapse"
aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarCollapse">
            <ul class="navbar-nav ml-auto">

                <li class="nav-item">
                    <a class="nav-link"
href="C:\Users\SLIM5\OneDrive\Documents\Project_sem6\HTML_Files\diseaseindex.h
tml" style="color: rgb(224, 227, 234);">Disease Index</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="http://192.168.0.104:8503"
style="color: rgb(224, 227, 234);">Breast Cancer</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="http://192.168.0.104:8502"
style="color: rgb(224, 227, 234);">Diabetes</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="http://192.168.0.104:8504"
style="color: rgb(224, 227, 234);">Heart Disease</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="http://192.168.0.104:8505"
style="color: rgb(224, 227, 234);">Parkinsons Disease</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="http://192.168.0.104:8501"
style="color: rgb(224, 227, 234);">Pneumonia Disease</a>
                </li>
            </ul>
        </div>
    </nav>

    <main role="main">

        <div class="jumbotron">
            <div class="container">
                <h1 class="display-4 text-center" style="color: white; background-
color: darkgreen; font-family: 'Times New Roman', Times, serif; font-weight:
bold;"><b>MyMedCare</b></h1>

                <p class="display-4 text-center" style="font-size: 1.5rem; color:
rgb(2, 49, 84);">"Welcome to our innovative platform dedicated to disease
prediction and personalized healthcare recommendations. At MyMedCare,

```

we utilize cutting-edge technology to forecast potential health risks and provide tailored suggestions for preventive measures.

Our advanced algorithms analyze various factors to predict the likelihood of specific diseases, empowering users to take proactive steps towards their well-being.

Additionally, our platform offers curated recommendations for doctors and hospitals based on individual needs, ensuring access to quality healthcare services. Join us in taking charge of your health journey today."</p>

```
</div>
</div>
<div>
    <p>
        <img src = "C:\Users\SLIM5\OneDrive\Documents\Project_sem6\New
folder\fullfledge.jpg" alt = "Image" class="img-fluid" >
    </p>
</div>
<div>
    <h4><p><marquee width="100%" behavior="scroll">
        <font face="Bahnschrift Condensed" size="10"
color="darkgreen">"Predict. Protect. Prevent"</font></marquee></p></h4>
        <p><marquee width="100%" behavior="scroll" direction="right">
            <font face="Bahnschrift Condensed" size="6"
color="darkblue">"Prediction and Recommendation for your Life"</font>
        </marquee></p>
    </div>
    </div>
    </div>
    <h2><div class="container">
        <div class="row">
            <div class="col-md-6">
                
            </div>
            <div class="col-md-6">
                <div class="about-us">
                    <h2>About Us</h2>
                    <p>Welcome to MyMedCare, your trusted partner in
healthcare.
                </div>
            </div>
        </div>
    </div>
    <h2>Our Mission</h2>
    <p>At MyMedCare, we believe in the power of technology to transform the way people approach healthcare. Our mission is to revolutionize the way you approach wellness by providing personalized, data-driven solutions. With a team of dedicated experts, we harness the power of technology to empower individuals to make informed decisions about their health. Through continuous innovation and research, we strive to stay at the forefront of preventive healthcare, ensuring you have access to the latest advancements in the field.
    </p>
</div>
```

Our mission is to revolutionize the way you approach wellness by providing personalized, data-driven solutions.

With a team of dedicated experts, we harness the power of technology to empower individuals to make informed decisions about their health.

Through continuous innovation and research, we strive to stay at the forefront of preventive healthcare, ensuring you have access to the latest advancements in the field.

```

At MyMedCare, we are committed to your well-being and
are here to support you every step of your health journey.</p>
<table>
  <tr>
    <td><h5 style="color: #1a237e; font-weight: bold; margin-bottom: 0;">PredictPreventProtect

```

Output:

MyMedCare

Disease Index Breast Cancer Diabetes Heart Disease Parkinsons Disease Pneumonia Disease

MyMedCare

"Welcome to our innovative platform dedicated to disease prediction and personalized healthcare recommendations. At MyMedCare, we utilize cutting-edge technology to forecast potential health risks and provide tailored suggestions for preventive measures. Our advanced algorithms analyze various factors to predict the likelihood of specific diseases, empowering users to take proactive steps towards their well-being. Additionally, our platform offers curated recommendations for doctors and hospitals based on individual needs, ensuring access to quality healthcare services. Join us in taking charge of your health journey today."

Diabetes Care PARKINSON'S DISEASE Heart Disease Breast Cancer Pneumonia Diabetes Care PARKINSON'S DISEASE Heart Disease Breast Cancer Pneumonia

Protect. Prevent"

"Prediction and Recommendation for your Life"

MyMedCare

Disease Index Breast Cancer Diabetes Heart Disease Parkinsons Disease Pneumonia Disease

"Predict. Protect. Prevent"

"Prediction and Recommendation for your Life"

About Us

Welcome to MyMedCare, your trusted partner in healthcare. Our mission is to revolutionize the way you approach wellness by providing personalized, data-driven solutions. With a team of dedicated experts, we harness the power of technology to empower individuals to make informed decisions about their health. Through continuous innovation and research, we strive to stay at the forefront of preventive healthcare, ensuring you have access to the latest advancements in the field. At MyMedCare, we are committed to your well-being and are here to support you every step of your health journey.

Predict

Over here a user can add its vitals and it will forecast the disease with its Potential Health Conditions and whether the patient is likely to be diagnosed with that disease. Diseases such as Breast Cancer, Pneumonia, Heart Disease, Diabetes, and Parkinson's Disease can be predicted here.

Protect

Here it will recommend the doctors according to the disease predicted. It will ask you whether you want to consult a Specialist or a Hospital with your Vital reports.

Protect

List common symptoms associated with each disease to help users identify potential health issues. Explain the causes or risk factors contributing to each disease.

Disease Index

- HTML document presents a Disease Index page structured with disease-specific information.
- **Document Type Declaration (`<!DOCTYPE html>`)** : Defines the document type and version of HTML being used.
- **HTML Root Element (`<html>`)** : The root element of the HTML document.
- **Head Section (`<head>`)** : Contains metadata about the HTML document, including character encoding, viewport settings, external stylesheets, and links to external resources.
- **Meta Tags**: Specify document metadata like character encoding (charset), viewport settings for responsive design, and links to external resources.
- **Bootstrap Stylesheet**: Importing Bootstrap CSS (bootstrap.min.css) from a CDN to apply predefined styles.
- **Canonical Link**: Defines the canonical URL for the document.
- **Custom CSS**: Inline CSS and an external stylesheet (index.css) are linked using Flask's url_for method to serve static files.
- **Body Section (`<body>`)** : Contains the visible content of the webpage.
- **Header (`<header>`)** : Typically used to contain introductory content at the top of the page.
- **Container (`<div class="container">`)** : Utilizes Bootstrap's grid system for responsive layout.
- **Heading (`<h1>, <h2>, <h3>, <h6>`)** : Various headings used to provide structure and hierarchy to the content.
- **Paragraphs (`<p>`)** : Used to display descriptive text about diseases, their symptoms, causes, treatment, and prevention.
- **Unordered Lists (`, `)** : Lists symptoms, causes, and prevention methods in bullet point format for readability.
- **Images (``)** : Display images related to each disease with descriptive captions.
- **Styling**: Inline styles (style attributes) are used to apply specific styles to elements such as text alignment, font colors, and image dimensions.

- **Script Tags:** Import JavaScript libraries (jQuery, Popper.js, Bootstrap JS) for enhanced interactivity and functionality.

The <head> section contains metadata such as character set (<meta charset="utf-8">) and viewport settings (<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">).

External resources like Bootstrap CSS (<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">) are linked to provide styling for the webpage.

Custom CSS (<link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='index.css') }}>) is included using Flask's url_for method to reference a local stylesheet.

Input:

This HTML template sets up the basic structure of a webpage, including necessary metadata, external stylesheets (Bootstrap), and internal styles for customizations. It provides a foundation for adding content and styling to create a complete webpage. The inclusion of Bootstrap ensures that the webpage will be responsive and styled according to Bootstrap's predefined CSS classes and components.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
    <link rel="canonical"
href="https://getbootstrap.com/docs/4.5/examples/carousel/">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='index.css') }}>
    <title>MyMedCare - Disease Index</title>
</head>
<style>
    #main-footer
    {
        color: #FFFFFF;
        font-family: "Segoe UI";
        background: #2B2B2B;
```

```

        text-align: center;
        margin-top: 75px;
        padding: 16px;
        bottom: 8px;
    }

    .img_holder img{
        max-width: 100%; max-height: 100%;
    }

    .btn-file {
        position: relative;
        overflow: hidden;
    }
    .btn-file input[type=file] {
        position: absolute;
        top: 0;
        right: 0;
        min-width: 100%;
        min-height: 100%;
        font-size: 100px;
        text-align: right;
        filter: alpha(opacity=0);
        opacity: 0;
        outline: none;
        background: white;
        cursor: inherit;
        display: block;
    }

```

</style>

<body>

This snippet demonstrates how to create a visually appealing section of a webpage using Bootstrap classes for layout and styling, along with inline styles for specific element customization. The content is centered within the container to improve readability and visual appeal.

```

<header>

</header>
<br/>
<br/>
<div class="container">
    <h1 class="display-4 text-center" style="color: white; background-
color: darkgreen; font-family: 'Times New Roman', Times, serif; font-weight:
bold;"><b>Disease Index</b></h1>

```

```
<h6 style="text-align:center;">Get all the information related to your disease here </h6>
<p style="text-align:center;">include a comprehensive list of diseases along with relevant information to help users understand each disease. It will Clearly list the name of each disease. Provide a brief description of each disease to give users an overview. List common symptoms associated with each disease to help users identify potential health issues. Explain the causes or risk factors contributing to each disease. </p>
```

this snippet demonstrates how HTML can be used to present structured information about a specific medical topic on a webpage. Each section is organized with appropriate headings and content to inform users about breast cancer, its symptoms, causes, treatment, and prevention.

```
<h2 style = "color : rgb(4, 95, 95)">1. Breast Cancer </h2>
<div style="text-align: center; margin-top: 20px">
    <br><br>
</div>
<h3>Overview</h3>
<p>A cancer that develops in the breast cells and progresses in stages. Few early symptoms may include new lump in the underarm or in breast, itching or discharge from the nipples, and skin texture change of the nipple or breast.<br></p>

<h3>Symptoms</h3>
<ul>
    <li>A lump or mass in the breast that feels different from the surrounding tissue</li>
    <li>Change in the size, shape or appearance of a breast.</li>
    <li>Discharge from the nipple</li>
    <li>Breast rash</li>
    <li>Changes in the skin over the breast, for example, dimpling</li>
    <li>Breast pain</li>
    <li>Inverted or pulling-in of the nipple</li>
    <li>Scaling, peeling, or flaking skin over the breast, particularly the dark area around the nipple</li>
    <li>Redness and/or pitting of the breast skin, resembling the skin of orange</li>
</ul>
<h3>Causes</h3>
<p>Doctors know that breast cancer occurs when some breast cells begin to grow abnormally. These cells divide more rapidly than healthy cells do and continue to accumulate, forming a lump or mass. Cells may spread (metastasize) through your breast to your lymph nodes or to other parts of your body.<br> Breast cancer most often begins with cells in the milk-producing ducts (invasive ductal carcinoma). Breast cancer may also begin in the glandular
```

tissue called lobules (invasive lobular carcinoma) or in other cells or tissue within the breast.

Researchers have identified hormonal, lifestyle and environmental factors that may increase your risk of breast cancer. But it's not clear why some people who have no risk factors develop cancer, yet other people with risk factors never do. It's likely that breast cancer is caused by a complex interaction of your genetic makeup and your environment.</p>

<h3>Treatment</h3>

- Treatment is primarily based on the type and stage of cancer.
- Uses high-powered energy rays to destroy cancer cells.- Can be done using a machine or by implanting a radioactive material inside the tumor mass
- Lumpectomy. The entire tumor mass and some of the surrounding healthy tissues are removed.

<h3 style = "color: rgb(21, 112, 21)">Prevention</h3>

<p>Limit or stay away from alcohol.

Stay at a healthy weight

Get active.

Breastfeed

Limit hormone therapy after menopause.</p>

<hr>

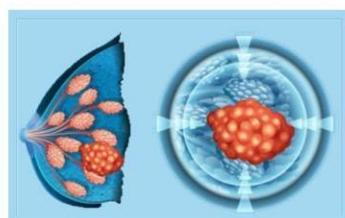
Output:

Disease Index

Get all the information related to your disease here

include a comprehensive list of diseases along with relevant information to help users understand each disease. It will Clearly list the name of each disease. Provide a brief description of each disease to give users an overview. List common symptoms associated with each disease to help users identify potential health issues. Explain the causes or risk factors contributing to each disease.

1. Breast Cancer



Overview

A cancer that develops in the breast cells and progresses in stages. Few early symptoms may include new lump in the underarm or in breast, itching or discharge from the nipples, and skin texture change of the nipple or breast.

Symptoms

- A lump or mass in the breast that feels different from the surrounding tissue
- Change in the size, shape or appearance of a breast.
- Discharge from the nipple
- Breast rash

- Changes in the skin over the breast, for example, dimpling
- Breast pain
- Inverted or pulling-in of the nipple
- Scaling, peeling, or flaking skin over the breast, particularly the dark area around the nipple
- Redness and/or pitting of the breast skin, resembling the skin of orange

Causes

Doctors know that breast cancer occurs when some breast cells begin to grow abnormally. These cells divide more rapidly than healthy cells do and continue to accumulate, forming a lump or mass. Cells may spread (metastasize) through your breast to your lymph nodes or to other parts of your body.

Breast cancer most often begins with cells in the milk-producing ducts (invasive ductal carcinoma). Breast cancer may also begin in the glandular tissue called lobules (invasive lobular carcinoma) or in other cells or tissue within the breast.

Researchers have identified hormonal, lifestyle and environmental factors that may increase your risk of breast cancer. But it's not clear why some people who have no risk factors develop cancer, yet other people with risk factors never do. It's likely that breast cancer is caused by a complex interaction of your genetic makeup and your environment.

Treatment

- Treatment is primarily based on the type and stage of cancer.
- Uses high-powered energy rays to destroy cancer cells.- Can be done using a machine or by implanting a radioactive material inside the tumor mass.
- Lumpectomy. The entire tumor mass and some of the surrounding healthy tissues are removed.

Prevention

Limit or stay away from alcohol.

Stay at a healthy weight

Get active.

Breastfeed

Limit hormone therapy after menopause.

this snippet demonstrates how HTML can be used to present structured information about another specific medical topic, diabetes, on a webpage. Each section is organized with appropriate headings and content to inform users about diabetes, its causes, symptoms, treatment, and prevention.

```
<h2 style = "color : rgb(4, 95, 95)">2. Diabetes</h2>
<div style="text-align: center; margin-top: 20px">
    <br/><br/>
</div>
<h3>Overview</h3>
<p>Diabetes is a disease that occurs when your blood glucose, also called blood sugar, is too high. Blood glucose is your main source of energy and comes from the food you eat. Insulin, a hormone made by the pancreas, helps glucose from food get into your cells to be used for energy.</p>
<h3>Causes</h3>
<ul>
    <li><b>Genetic Predisposition:</b> Individuals with a family history of diabetes are at higher risk due to inherited genetic susceptibility.</li>
    <li><b>Lifestyle Choices:</b> Sedentary lifestyles, poor dietary habits, and obesity significantly increase the likelihood of developing type 2 diabetes. Consuming excessive refined sugars, unhealthy fats, and processed foods can contribute to insulin resistance and weight gain.</li>
    <li><b>Physical Inactivity:</b> Lack of regular physical activity impairs glucose metabolism and insulin sensitivity, predisposing individuals to insulin resistance and type 2 diabetes.</li>

```

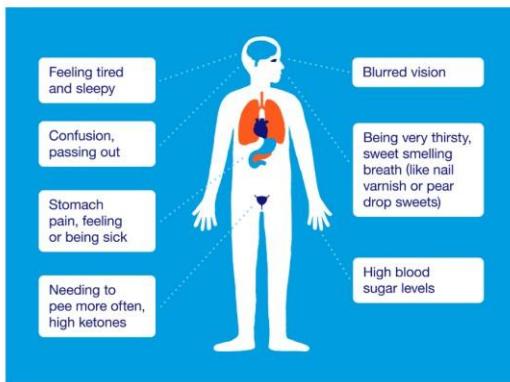
```
<li><b>Gestational Factors:</b> Gestational diabetes, which occurs during pregnancy, can increase the risk of developing type 2 diabetes later in life for both the mother and the child.</li>
</ul>

<h3>Symptoms</h3>
<ul>
<li>Feeling more thirsty than usual.</li>
<li>Urinating often.</li>
<li>Losing weight without trying</li>
<li>Presence of ketones in the urine. Ketones are a byproduct of the breakdown of muscle and fat that happens when there's not enough available insulin.</li>
<li>Feeling tired and weak.</li>
<li>Feeling irritable or having other mood changes.</li>
<li>Having blurry vision.</li>
<li>Having slow-healing sores.</li>
<li>Getting a lot of infections, such as gum, skin and vaginal infections.</li>
</ul>

<h3>Treatment</h3>
<p>Doctors treat diabetes with a few different medications. Some are taken by mouth, while others are available as injections. Doctors treat diabetes with a few different medications. Some are taken by mouth, while others are available as injections. Insulin is the main treatment for type 1 and 1.5 diabetes. It replaces the hormone your body isn't able to produce.</p>
<br>
<h3 style = "color:  rgb(21, 112, 21)">Prevention</h3>
<p>Lose extra weight<br>
    Be more physically active<br>
    Eat healthy plant foods<br>
    Eat healthy fats<br>
    Skip fad diets and make healthier choices</p>
<hr>
<br>
```

Output:

2. Diabetes



Overview

Diabetes is a disease that occurs when your blood glucose, also called blood sugar, is too high. Blood glucose is your main source of energy and comes from the food you eat. Insulin, a hormone made by the pancreas, helps glucose from food get into your cells to be used for energy.

Causes

- **Genetic Predisposition:** Individuals with a family history of diabetes are at higher risk due to inherited genetic susceptibility.
- **Lifestyle Choices:** Sedentary lifestyles, poor dietary habits, and obesity significantly increase the likelihood of developing type 2 diabetes. Consuming excessive refined sugars, unhealthy fats, and processed foods can contribute to insulin resistance and weight gain.
- **Physical Inactivity:** Lack of regular physical activity impairs glucose metabolism and insulin sensitivity, predisposing individuals to insulin resistance
- **Gestational Factors:** Gestational diabetes, which occurs during pregnancy, can increase the risk of developing type 2 diabetes later in life for both the mother and the child.

Symptoms

- Feeling more thirsty than usual.
- Urinating often.
- Losing weight without trying
- Presence of ketones in the urine. Ketones are a byproduct of the breakdown of muscle and fat that happens when there's not enough available insulin.
- Feeling tired and weak.
- Feeling irritable or having other mood changes.
- Having blurry vision.
- Having slow-healing sores.
- Getting a lot of infections, such as gum, skin and vaginal infections.

Treatment

Doctors treat diabetes with a few different medications. Some are taken by mouth, while others are available as injections. Doctors treat diabetes with a few different medications. Some are taken by mouth, while others are available as injections. Insulin is the main treatment for type 1 and 1.5 diabetes. It replaces the hormone your body isn't able to produce.

Prevention

Lose extra weight
Be more physically active
Eat healthy plant foods
Eat healthy fats
Skip fad diets and make healthier choices

snippet demonstrates how HTML can be used to present structured information about another specific medical topic, heart disease, on a webpage. Each section is organized with appropriate headings and content to inform users about heart disease, its symptoms, causes, treatment, and prevention.

```
<h2 style = "color : rgb(4, 95, 95)">3.Heart Disease</h2>
<div style="text-align: center; margin-top: 20px">
```

```
        <br/>  
    </div>  
<h3>Overview</h3>  
<p>Heart disease describes a range of conditions that affect your heart. Diseases under the heart disease umbrella include blood vessel diseases, such as coronary artery disease; heart rhythm problems (arrhythmias); and heart defects you're born with (congenital heart defects), among others.<br>  
The term "heart disease" is often used interchangeably with the term "cardiovascular disease." Cardiovascular disease generally refers to conditions that involve narrowed or blocked blood vessels that can lead to a heart attack, chest pain (angina) or stroke. Other heart conditions, such as those that affect your heart's muscle, valves or rhythm, also are considered forms of heart disease.<br>  
Many forms of heart disease can be prevented or treated with healthy lifestyle choices.</p>  
  
<h3>Symptoms</h3>  
<ul>  
    <li>Chest pain, chest tightness, chest pressure and chest discomfort (angina)</li>  
    <li>Shortness of breath</li>  
    <li>Pain, numbness, weakness or coldness in your legs or arms if the blood vessels in those parts of your body are narrowed</li>  
    <li>Pain in the neck, jaw, throat, upper abdomen or back</li>  
</ul>  
<h3>Causes</h3>  
<ul>  
    <li>Elevated blood pressure puts strain on the heart and blood vessels, increasing the risk of heart disease, heart attack, and stroke.</li>  
    <li>Elevated levels of LDL cholesterol (often referred to as "bad" cholesterol) can lead to the buildup of plaque in the arteries, narrowing them and restricting blood flow to the heart, which may result in heart disease and heart attacks.</li>  
    <li>Tobacco use, including smoking and exposure to secondhand smoke, significantly increases the risk of heart disease by damaging blood vessels, reducing oxygen in the blood, and raising blood pressure.</li>  
    <li>Diets high in saturated fats, trans fats, cholesterol, salt (sodium), and sugar increase the risk of heart disease. Conversely, diets rich in fruits, vegetables, whole grains, and lean proteins can help reduce the risk.</li>  
    <li>People with diabetes are at higher risk of developing heart disease due to factors such as high blood sugar levels, insulin resistance, and other metabolic abnormalities that affect the cardiovascular system.</li>  
    <li>Chronic stress and untreated mental health conditions like depression and anxiety can contribute to heart disease by increasing blood pressure,</li>
```

promoting unhealthy coping behaviors (such as smoking or overeating), and triggering inflammation in the body.

The risk of heart disease increases with age. As people get older, their risk of developing heart disease, especially coronary artery disease, rises due to factors such as decreased elasticity of blood vessels and accumulation of plaque.

<h3>Treatment</h3>

<p>Adopting a heart-healthy lifestyle is crucial for managing heart disease. This includes maintaining a healthy diet low in saturated fats, trans fats, cholesterol, and sodium; engaging in regular physical activity; quitting smoking; and managing stress.

Angioplasty and stenting to open blocked or narrowed coronary arteries.

Cardiac rehabilitation programs offer comprehensive, supervised exercise training, education, and support to improve cardiovascular health and reduce the risk of future heart events. These programs are typically recommended after a heart attack, heart surgery, or for individuals with stable heart disease.

Regular monitoring of heart health through tests such as electrocardiograms (ECGs or EKGs), echocardiograms, stress tests, and blood tests is essential for managing heart disease and adjusting treatment as needed.

</p>

<h3 style = "color: rgb(21, 112, 21)">Prevention</h3>

<p>Don't smoke or use tobacco

Get moving: Aim for at least 30 to 60 minutes of activity daily

Eat a heart-healthy diet

Maintain a healthy weight

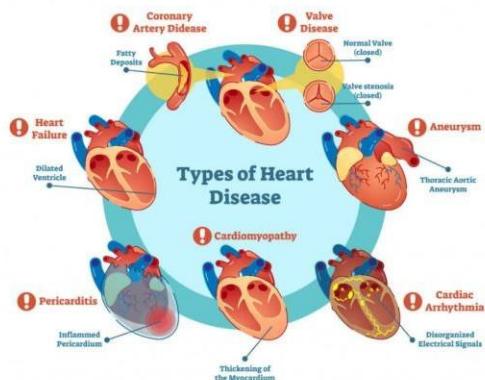
Get quality sleep

Manage stress

Get regular health screening tests</p>

Output:

3. Heart Disease



Overview

Heart disease describes a range of conditions that affect your heart. Diseases under the heart disease umbrella include blood vessel diseases, such as coronary artery disease; heart rhythm problems (arrhythmias); and heart defects you're born with (congenital heart defects), among others.

The term "heart disease" is often used interchangeably with the term "cardiovascular disease." Cardiovascular disease generally refers to conditions that involve narrowed or blocked blood vessels that can lead to a heart attack, chest pain (angina) or stroke. Other heart conditions, such as those that affect your heart's muscle, valves or rhythm, also are considered forms of heart disease.

Many forms of heart disease can be prevented or treated with healthy lifestyle choices.

Symptoms

- Chest pain, chest tightness, chest pressure and chest discomfort (angina)
- Shortness of breath
- Pain, numbness, weakness or coldness in your legs or arms if the blood vessels in those parts of your body are narrowed
- Pain in the neck, jaw, throat, upper abdomen or back

Causes

- Elevated blood pressure puts strain on the heart and blood vessels, increasing the risk of heart disease, heart attack, and stroke.
- Elevated levels of LDL cholesterol (often referred to as "bad" cholesterol) can lead to the buildup of plaque in the arteries, narrowing them and restricting blood flow to the heart, which may result in heart disease and heart attacks.
- Tobacco use, including smoking and exposure to secondhand smoke, significantly increases the risk of heart disease by damaging blood vessels, reducing oxygen in the blood, and raising blood pressure.
- Diets high in saturated fats, trans fats, cholesterol, salt (sodium), and sugar increase the risk of heart disease. Conversely, diets rich in fruits, vegetables, whole grains, and lean proteins can help reduce the risk.
- People with diabetes are at higher risk of developing heart disease due to factors such as high blood sugar levels, insulin resistance, and other metabolic abnormalities that affect the cardiovascular system.
- Chronic stress and untreated mental health conditions like depression and anxiety can contribute to heart disease by increasing blood pressure, promoting unhealthy coping behaviors (such as smoking or overeating), and triggering inflammation in the body.
- The risk of heart disease increases with age. As people get older, their risk of developing heart disease, especially coronary artery disease, rises due to factors such as decreased elasticity of blood vessels and accumulation of plaque.

Treatment

Adopting a heart-healthy lifestyle is crucial for managing heart disease. This includes maintaining a healthy diet low in saturated fats, trans fats, cholesterol, and sodium; engaging in regular physical activity; quitting smoking; and managing stress.

Angioplasty and stenting to open blocked or narrowed coronary arteries.

Cardiac rehabilitation programs offer comprehensive, supervised exercise training, education, and support to improve cardiovascular health and reduce the risk of future heart events. These programs are typically recommended after a heart attack, heart surgery, or for individuals with stable heart disease.

Regular monitoring of heart health through tests such as electrocardiograms (ECGs or EKGs), echocardiograms, stress tests, and blood tests is essential for managing heart disease and adjusting treatment as needed.

Prevention

Don't smoke or use tobacco

Get moving: Aim for at least 30 to 60 minutes of activity daily

Eat a heart-healthy diet

Maintain a healthy weight

Get quality sleep

Manage stress

Get regular health screening tests

HTML code snippet provides detailed information about Parkinson's Disease, covering its overview, causes, symptoms, treatment, and prevention.

```
<hr>
<br>
<h2 style = "color : rgb(4, 95, 95)">4. Parkinsons Disease</h2>
<div style="text-align: center; margin-top: 20px">
    <br/>
</div>
<h3>Overview</h3>
<p>Parkinson disease (PD) is a brain condition that causes problems with movement, mental health, sleep, pain and other health issues.
    PD gets worse over time. There is no cure, but therapies and medicines can reduce symptoms. Common symptoms include tremors, painful muscle contractions and difficulty speaking.
    Parkinson disease results in high rates of disability and the need for care. Many people with PD also develop dementia.
    The disease usually occurs in older people, but younger people can also be affected. Men are affected more often than women.
    The cause of PD is unknown but people with a family history of the disease have a higher risk. Exposure to air pollution, pesticides and solvents may increase risk.
</p>
<h3>Causes</h3>
<ul>
    <li>Parkinson's disease, certain nerve cells called neurons in the brain gradually break down or die. Many of the symptoms of Parkinson's are due to a loss of neurons that produce a chemical messenger in your brain called dopamine. When dopamine levels decrease, it causes irregular brain activity, leading to problems with movement and other symptoms of Parkinson's disease.</li>
    <li>Clumps of specific substances within brain cells are microscopic markers of Parkinson's disease. These are called Lewy bodies, and researchers believe these Lewy bodies hold an important clue to the cause of Parkinson's disease.</li>
    <li>Exposure to certain toxins or environmental factors may increase the risk of later Parkinson's disease, but the risk is small.</li>
    <li>Men are more likely to develop Parkinson's disease than are women.</li>
    <li>Many of the symptoms of Parkinson's are due to a loss of neurons that produce a chemical messenger in your brain called dopamine. When dopamine levels decrease, it causes irregular brain activity, leading to problems with movement and other symptoms of Parkinson's disease.</li>
</ul>
<h3>Symptoms</h3>
<ul>
```

```
<li>tremor</li>
<li>involuntary movement</li>
<li>rigidity</li>
<li>trouble walking</li>
<li>imbalance</li>
<li>cognitive impairment</li>
<li>mental health disorders</li>
<li>dementia</li>
<li>sleep disorders</li>
<li>pain</li>
<li>sensory disturbances</li>
</ul>
<h3>Treatment</h3>
<p>There is no cure for Parkinson disease, but therapies including medicines, surgery and rehabilitation can reduce symptoms.

Levodopa/carbidopa, a combination medicine that increases the amount of dopamine in the brain, is the most common medication for PD (1). Doctors may use other medicines such as anticholinergics to reduce involuntary muscle movement.

Deep brain stimulation and other therapies can help the tremors and reduce the need for medicines.

Rehabilitation including physiotherapy can offer relief for Parkinson disease</p>
<br>
<h3 style = "color:  rgb(21, 112, 21)">Prevention</h3>
<p>Because the cause of Parkinson's is unknown, there are no proven ways to prevent the disease.

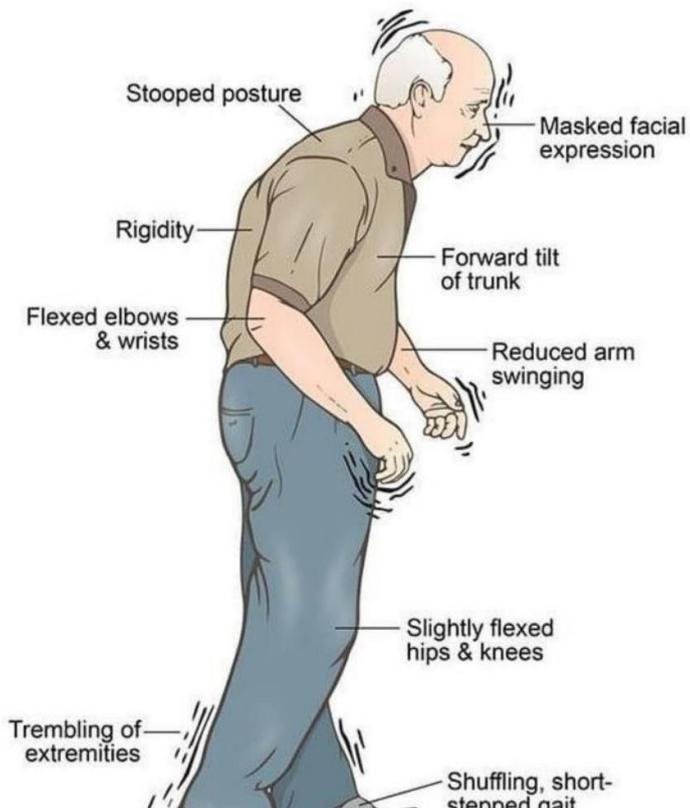
Some research has shown that regular aerobic exercise might reduce the risk of Parkinson's disease.

Some other research has shown that people who consume caffeine – which is found in coffee, tea and cola – get Parkinson's disease less often than those who don't drink it. Green tea also is related to a reduced risk of developing Parkinson's disease. However, it is still not known whether caffeine protects against getting Parkinson's or is related in some other way. Currently there is not enough evidence to suggest that drinking caffeinated beverages protects against Parkinson's.</p>
```

Output:

4. Parkinsons Disease

Typical appearance of Parkinson's disease



Overview

Parkinson disease (PD) is a brain condition that causes problems with movement, mental health, sleep, pain and other health issues. PD gets worse over time. There is no cure, but therapies and medicines can reduce symptoms. Common symptoms include tremors, painful muscle contractions and difficulty speaking. Parkinson disease results in high rates of disability and the need for care. Many people with PD also develop dementia. The disease usually occurs in older people, but younger people can also be affected. Men are affected more often than women. The cause of PD is unknown but people with a family history of the disease have a higher risk. Exposure to air pollution, pesticides and solvents may increase risk.

Causes

- Parkinson's disease, certain nerve cells called neurons in the brain gradually break down or die. Many of the symptoms of Parkinson's are due to a loss of neurons that produce a chemical messenger in your brain called dopamine. When dopamine levels decrease, it causes irregular brain activity, leading to problems with movement and other symptoms of Parkinson's disease.
- Clumps of specific substances within brain cells are microscopic markers of Parkinson's disease. These are called Lewy bodies, and researchers believe these Lewy bodies hold an important clue to the cause of Parkinson's disease.
- Exposure to certain toxins or environmental factors may increase the risk of later Parkinson's disease, but the risk is small.
- Men are more likely to develop Parkinson's disease than are women.
- Many of the symptoms of Parkinson's are due to a loss of neurons that produce a chemical messenger in your brain called dopamine. When dopamine levels decrease, it causes irregular brain activity, leading to problems with movement and other symptoms of Parkinson's disease.

Symptoms

- tremor
- involuntary movement
- rigidity
- trouble walking
- imbalance
- cognitive impairment
- mental health disorders
- dementia
- sleep disorders
- pain
- sensory disturbances

Treatment

There is no cure for Parkinson disease, but therapies including medicines, surgery and rehabilitation can reduce symptoms. Levodopa/carbidopa, a combination medicine that increases the amount of dopamine in the brain, is the most common medication for PD (1). Doctors may use other medicines such as anticholinergics to reduce involuntary muscle movement. Deep brain stimulation and other therapies can help the tremors and reduce the need for medicines. Rehabilitation including physiotherapy can offer relief for Parkinson disease

Prevention

Because the cause of Parkinson's is unknown, there are no proven ways to prevent the disease. Some research has shown that regular aerobic exercise might reduce the risk of Parkinson's disease. Some other research has shown that people who consume caffeine — which is found in coffee, tea and cola — get Parkinson's disease less often than those who don't drink it. Green tea also is related to a reduced risk of developing Parkinson's disease. However, it is still not known whether caffeine protects against getting Parkinson's or is related in some other way. Currently there is not enough evidence to suggest that drinking caffeinated beverages protects against Parkinson's.

HTML structure organizes content into sections, utilizing headings, paragraphs, lists, and images to present comprehensive information about Pneumonia Disease. The styling attributes (style="...") applied directly to HTML elements control visual aspects like text color, alignment, and image size, enhancing the readability and presentation of the content on a webpage.

```
<hr>
<br>
<h2 style = "color : rgb(4, 95, 95)">5. Pneumonia Disease</h2>
<div style="text-align: center; margin-top: 20px">
    <br/>
</div>
<h3>Overview</h3>
<p>Pneumonia is an infection that inflames the air sacs in one or both lungs. The air sacs may fill with fluid or pus (purulent material), causing cough with phlegm or pus, fever, chills, and difficulty breathing. A variety of organisms, including bacteria, viruses and fungi, can cause pneumonia.<br>
    Pneumonia can range in seriousness from mild to life-threatening. It is most serious for infants and young children, people older than age 65, and people with health problems or weakened immune systems.</p>

<h3>Symptoms</h3>
<ul>
    <li>Chest pain when you breathe or cough</li>
    <li>Confusion or changes in mental awareness (in adults age 65 and older)</li>
    <li>Fatigue</li>
    <li>Fever, sweating and shaking chills</li>
    <li>Nausea, vomiting or diarrhea</li>
    <li>Shortness of breath</li>
</ul>
```

```
<br>

<h3>Causes</h3>
<ul>
    <li>The most common cause of bacterial pneumonia in the U.S. is Streptococcus pneumoniae. This type of pneumonia can occur on its own or after you've had a cold or the flu. It may affect one part (lobe) of the lung, a condition called lobar pneumonia.</li>
    <li>Viruses, including COVID-19. Some of the viruses that cause colds and the flu can cause pneumonia. Viruses are the most common cause of pneumonia in children younger than 5 years. Viral pneumonia is usually mild. But in some cases it can become very serious. Coronavirus 2019 (COVID-19) may cause pneumonia, which can become severe.</li>
    <li>This type of pneumonia is most common in people with chronic health problems or weakened immune systems, and in people who have inhaled large doses of the organisms.</li>
    <li>Smoking damages your body's natural defenses against the bacteria and viruses that cause pneumonia.</li>
</ul>

<h3>Treatment</h3>
<p>Treatment for pneumonia involves curing the infection and preventing complications. People who have community-acquired pneumonia usually can be treated at home with medication. Although most symptoms ease in a few days or weeks, the feeling of tiredness can persist for a month or more.<br>
    You may take these as needed for fever and discomfort. These include drugs such as aspirin, ibuprofen (Advil, Motrin IB, others) and acetaminophen (Tylenol, others).<br>
    These medicines are used to treat bacterial pneumonia. It may take time to identify the type of bacteria causing your pneumonia and to choose the best antibiotic to treat it. If your symptoms don't improve, your doctor may recommend a different antibiotic</p>

<h3 style = "color:  rgb(21, 112, 21)">Prevention</h3>
<p>Get vaccinated.<br>
    Practice good hygiene<br>
    Don't smoke. <br>
    Keep your immune system strong.</p>
<hr>

</div>
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCKR/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
" integrity="sha384-
```

```
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous">></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-
JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQQxSFFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous">></script>
</body>
</html>
```

Output:

5. Pneumonia Disease



Overview

Pneumonia is an infection that inflames the air sacs in one or both lungs. The air sacs may fill with fluid or pus (purulent material), causing cough with phlegm or pus, fever, chills, and difficulty breathing. A variety of organisms, including bacteria, viruses and fungi, can cause pneumonia. Pneumonia can range in seriousness from mild to life-threatening. It is most serious for infants and young children, people older than age 65, and people with health problems or weakened immune systems.

Symptoms

- Chest pain when you breathe or cough
- Confusion or changes in mental awareness (in adults age 65 and older)
- Fatigue
- Fever, sweating and shaking chills
- Nausea, vomiting or diarrhea
- Shortness of breath

Causes

- The most common cause of bacterial pneumonia in the U.S. is Streptococcus pneumoniae. This type of pneumonia can occur on its own or after you've had a cold or the flu. It may affect one part (lobe) of the lung, a condition called lobar pneumonia.
- Viruses, including COVID-19. Some of the viruses that cause colds and the flu can cause pneumonia. Viruses are the most common cause of pneumonia in children younger than 5 years. Viral pneumonia is usually mild. But in some cases it can become very serious. Coronavirus 2019 (COVID-19) may cause pneumonia, which can become severe.
- This type of pneumonia is most common in people with chronic health problems or weakened immune systems, and in people who have inhaled large doses of the organisms.
- Smoking damages your body's natural defenses against the bacteria and viruses that cause pneumonia.

Treatment

Treatment for pneumonia involves curing the infection and preventing complications. People who have community-acquired pneumonia usually can be treated at home with medication. Although most symptoms ease in a few days or weeks, the feeling of tiredness can persist for a month or more. You may take these as needed for fever and discomfort. These include drugs such as aspirin, ibuprofen (Advil, Motrin IB, others) and acetaminophen (Tylenol, others). These medicines are used to treat bacterial pneumonia. It may take time to identify the type of bacteria causing your pneumonia and to choose the best antibiotic to treat it. If your symptoms don't improve, your doctor may recommend a different antibiotic.

Prevention

Get vaccinated.
Practice good hygiene
Don't smoke.
Keep your immune system strong.

Breast Cancer Prediction

Predicting breast cancer involves assessing various risk factors and utilizing medical imaging and diagnostic tests. While a definitive prediction of breast cancer cannot be made without proper medical evaluation.

Backend

```
# In[1]:  
  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
from sklearn.metrics import classification_report  
from sklearn.metrics import accuracy_score  
from sklearn.metrics import confusion_matrix  
from sklearn.model_selection import train_test_split  
from sklearn.model_selection import cross_val_score  
from sklearn.model_selection import KFold  
from sklearn.tree import DecisionTreeClassifier  
from sklearn.neighbors import KNeighborsClassifier  
from sklearn.naive_bayes import GaussianNB  
from sklearn.pipeline import Pipeline  
from sklearn.preprocessing import StandardScaler  
from sklearn.model_selection import GridSearchCV  
from sklearn.svm import SVC  
from pandas.plotting import scatter_matrix  
import seaborn as sns
```

This Python environment sets up essential libraries for machine learning tasks related to breast cancer prediction. It includes tools for data manipulation and analysis using Pandas (`pd`), data visualization with Matplotlib (`plt`) and Seaborn (`sns`), and model building using algorithms like Decision Trees (`DecisionTreeClassifier`), K-Nearest Neighbors (`KNeighborsClassifier`), Naive Bayes (`GaussianNB`), and Support Vector Machines (`SVC`). You can perform model evaluation using metrics such as accuracy, confusion matrix, and classification report (`classification_report`) along with techniques like cross-validation (`cross_val_score`) and hyperparameter tuning (`GridSearchCV`). StandardScaler (`StandardScaler`) is available for feature scaling, and pipeline construction (`Pipeline`) facilitates efficient data preprocessing and modeling. This setup enables comprehensive analysis and prediction of breast cancer based on provided datasets.

```
# In[2]:  
  
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/breast-  
cancer-wisconsin/breast-cancer-wisconsin.data"  
names = ['id', 'clump_thickness', 'uniform_cell_size', 'uniform_cell_shape',  
        'marginal_adhesion', 'single_epithelial_size', 'bare_nuclei',  
        'bland_chromatin', 'normal_nucleoli', 'mitoses', 'class']  
df = pd.read_csv(url, names=names)
```

a dataset related to breast cancer from the UCI Machine Learning Repository is loaded into a Pandas DataFrame (`df`). The dataset is sourced from the URL specified (`url`). Column names for the dataset are defined in the `names` list, which includes attributes such as 'id', 'clump_thickness', 'uniform_cell_size', 'uniform_cell_shape', 'marginal_adhesion', 'single_epithelial_size', 'bare_nuclei', 'bland_chromatin', 'normal_nucleoli', 'mitoses', and 'class'. These attributes represent various features associated with breast cancer cell characteristics. The dataset is read into `df` using `pd.read_csv()`, ensuring that each column is labeled appropriately for subsequent analysis and modeling tasks.

```
# In[3]:  
  
df.head()
```

uniform_cell_shape	marginal_adhesion	single_epithelial_size	bare_nuclei	bland_chromatin	normal_nucleoli	mitoses	class
1	1	2	1	3	1	1	2
4	5	7	10	3	2	1	2
1	1	2	2	3	1	1	2
8	1	3	4	3	7	1	2
1	3	2	1	3	1	1	2

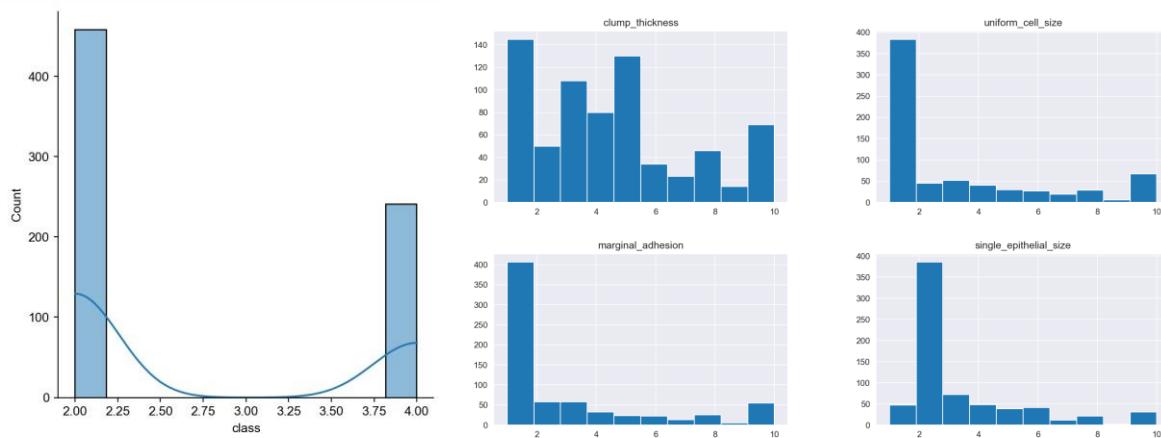
These steps collectively prepare the dataset for further analysis and machine learning tasks by ensuring that irrelevant columns are removed and missing values are appropriately handled through imputation.

```
# In[4]:  
#Shape of the Dataset  
df.shape  
# ## Data Pre-processing  
df.drop(['id'],axis=1,inplace = True)  
# Columns in the dataset  
df.columns  
df.info()  
df['class'].value_counts()  
df['bare_nuclei'].value_counts()  
df[df['bare_nuclei'] == '?'].sum()  
df.replace('?',np.nan,inplace=True)  
df['bare_nuclei'][23]  
df.isna().sum()  
df.fillna(method='ffill', inplace=True)  
df.isna().sum()
```

These commands offer a comprehensive overview of the dataset. The info() method provides details on the data types, non-null values, and memory usage of each column. Meanwhile, describe() generates descriptive statistics that summarize the central tendency, dispersion, and shape of numerical features.

```
# ## Exploratory Data Analysis  
  
df.info()  
df.describe()  
  
# ## Bivariate Data Analysis  
sns.displot(df['class'],kde=True)  
  
# ## Multivariate Data Analysis  
  
sns.set_style('darkgrid')  
df.hist(figsize=(30,30))  
plt.show()  
  
plt.figure(figsize=(10,10))  
sns.boxplot(data=df,orient='h')
```

Output:



```
# ## Feature Selection  
  
df.corr()  
  
plt.figure(figsize=(30,20))  
cor = df.corr()  
sns.heatmap(cor,vmax=1,square = True,annot=True, cmap=plt.cm.Blues)  
plt.title('Correlation between different attributes')  
plt.show()  
  
sns.pairplot(df,diag_kind='kde')
```

These visualizations aid in identifying potential correlations and patterns within the dataset, guiding feature selection and further analysis for predictive modeling. The heatmap highlights attribute relationships, while the pairplot provides insights into variable distributions and potential dependencies. These are crucial steps in understanding the dataset's underlying structure and informing subsequent machine learning tasks.

```
#Correlation with output variable  
cor_target = abs(cor["class"])  
#Selecting highly correlated features  
relevant_features = cor_target[cor_target>0]  
relevant_features
```

The code snippet computes the absolute correlation coefficients between the "class" variable (output variable) and all other features in the breast cancer dataset (df). The cor_target

variable stores these correlation values, and then relevant features with correlations greater than 0 are selected using `cor_target[cor_target>0]`.

This approach identifies the features that have a significant correlation with the target variable ("class"), which is important for feature selection in predictive modeling tasks. By focusing on highly correlated features, you can prioritize those that potentially contribute more to predicting the target variable, thus improving the effectiveness of the machine learning model.

```
# ## Training and Testing

# In[22]:


Y = df['class'].values
X = df.drop('class', axis=1).values

X_train, X_test, Y_train, Y_test = train_test_split (X, Y, test_size = 0.30,
random_state=21)

# In[23]:


scoring = 'accuracy'

# In[24]:


models= []
models.append((‘CART’, DecisionTreeClassifier()))
models.append((‘SVM’, SVC()))
models.append((‘NB’, GaussianNB()))
models.append((‘KNN’, KNeighborsClassifier()))

# evaluate each model in turn
results = []
names = []

for name, model in models:
    kfold = KFold(n_splits=10)
    cv_results = cross_val_score(model, X_train, Y_train, cv=kfold,
scoring=scoring)
    results.append(cv_results)
    names.append(name)
    msg = "For %s Model:Mean accuracy is %f (Std accuracy is %f)" % (name,
cv_results.mean(), cv_results.std())
    print(msg)
```

```
# In[25]:  
  
fig = plt.figure(figsize=(10,10))  
fig.suptitle('Performance Comparison')  
ax = fig.add_subplot(111)  
plt.boxplot(results)  
ax.set_xticklabels(names)  
plt.show()
```

The code performs the training and testing of various machine learning models using k-fold cross-validation. It splits the dataset into training and testing sets, with 70% of the data used for training and 30% for testing. Four different models are evaluated: Decision Tree (CART), Support Vector Machine (SVM), Naive Bayes (NB), and K-Nearest Neighbors (KNN). For each model, the code computes the mean accuracy and standard deviation of accuracy across 10 folds of cross-validation on the training data. Finally, it visualizes the performance comparison of these models using a boxplot, showing the distribution of accuracy scores obtained from cross-validation.

```
# Make predictions on validation dataset  
  
for name, model in models:  
    model.fit(X_train, Y_train)  
    predictions = model.predict(X_test)  
    print("\nModel:",name)  
    print("Accuracy score:",accuracy_score(Y_test, predictions))  
    print("Classification report:\n",classification_report(Y_test,  
predictions))
```

This code snippet makes predictions on the validation dataset using each of the machine learning models trained earlier. It iterates through the list of models, fits each model to the training data (`X_train`, `Y_train`), and then uses the trained model to predict the target variable (`Y_test`) based on the features (`X_test`). For each model, it prints the accuracy score and the classification report, which includes metrics such as precision, recall, F1-score, and support for each class in the target variable.

This process evaluates how well each model performs on the unseen validation data (`X_test`, `Y_test`) and provides insights into the model's predictive capabilities and performance metrics.

```

# ## Support Vector Machine

# In[27]:


clf = SVC()

clf.fit(X_train, Y_train)
accuracy = clf.score(X_test, Y_test)
print("Test Accuracy:",accuracy)

predict = clf.predict(X_test)
predict


# In[28]:


example_measures = [[4,2,1,1,1,2,3,2,1]]
prediction = clf.predict(example_measures)
print(prediction)


# In[29]:


import itertools
sns.set_theme(style="dark")
def plot_confusion_matrix(cm, classes, normalize=False,title='Confusion matrix', cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

    print(cm)

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

```

```

fmt = '.2f' if normalize else 'd'
thresh = cm.max() / 2.
for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
    plt.text(j, i, format(cm[i, j], fmt),
              horizontalalignment="center",
              color="white" if cm[i, j] > thresh else "black")

plt.tight_layout()
plt.ylabel('True label')
plt.xlabel('Predicted label')

# In[30]:


cnf_matrix = confusion_matrix(Y_test, predict, labels=[2,4])
np.set_printoptions(precision=2)

print (classification_report(Y_test, predict))

# Plot non-normalized confusion matrix
plt.figure()
plot_confusion_matrix(cnf_matrix,
classes=['Benign(2)', 'Malignant(4)'], normalize= False, title='Confusion matrix')


# In[31]:


import pickle
pickle.dump(clf, open('model.pkl','wb'))

model = pickle.load(open('model.pkl','rb'))
print(model.predict([[4,2,1,1,1,2,3,2,1]]))

```

The dataset is loaded from a URL into a pandas DataFrame. Initial preprocessing steps include dropping irrelevant columns (id), handling missing values (replacing '?' with NaN and forward filling), and preparing the feature set (X) and target variable (Y) for modeling.

Various aspects of the dataset are explored using descriptive statistics (info() and describe()), visualizations (histograms and boxplots), and correlation analysis (corr() and heatmap).

Four machine learning models (Decision Tree, SVM, Naive Bayes, K-Nearest Neighbors) are evaluated using k-fold cross-validation to assess their mean accuracy and standard deviation. The performance of these models is compared using boxplots.

Each model is trained on the training dataset (X_{train} , Y_{train}) and evaluated on the test dataset (X_{test} , Y_{test}). Predictions are made on the test dataset, and performance metrics (accuracy score, classification report) are computed and displayed for each model.

A specific focus on the SVM model is demonstrated, including fitting the model to the training data, evaluating test accuracy, making predictions on new examples, generating a confusion matrix, and finally saving/loading the trained model using pickle.

Interface

a Streamlit app for breast cancer prediction using the model you trained earlier. This app allows users to input various parameters related to breast cancer and provides a prediction based on the trained model.

```
import streamlit as st
import pickle
import numpy as np

def load_model():
    return
pickle.load(open("C:/Users/SLIM5/OneDrive/Documents/Project_sem6/Python_files/
SAV files/model.pkl", 'rb'))

def predict_breastcancer(input_data):
    model = load_model()
    prediction = model.predict(input_data)
    return prediction[0]

st.markdown("""
<style>
/* Center-align the title */
.title {
    text-align: center;
}
/* Style the title */
.title h1 {
    background-color: green;
    color: white;
    font-family: 'Times New Roman', Times, serif;
}
/* Set background image */
body {
    background-image:
url('C:/Users/SLIM5/OneDrive/Documents/Project_sem6/New folder/okay.jpg');
    background-size: cover;
```

```

        }
    </style>
"""
unsafe_allow_html=True)

# Title
def main():
    st.markdown("<div class='title'><h1>Breast Cancer
Prediction</h1></div>", unsafe_allow_html=True)
    # Description
    st.markdown("<p style='text-align: center; font-weight: bold;'>Enter
Patients Detail.</p>", unsafe_allow_html=True)

    # Form-like layout with input fields and placeholders
    col1, col2 = st.columns(2)

```

- **Model Loading and Prediction:** You have functions to load the trained model (load_model) and make predictions (predict_breastcancer) based on input data.
- **Streamlit Layout:** You've created a Streamlit layout using columns (st.columns) to organize input fields for different parameters related to breast cancer. Each input field corresponds to a specific feature used in your machine learning model.
- **User Input and Prediction:** Users can input values for different features such as clump thickness, uniform cell size, epithelial cell size, etc. After entering the required details, they can click a button (Predict Breast Cancer) to trigger the prediction based on the model.
- **Display of Prediction:** Depending on the prediction result (high or low risk of breast cancer), the app displays a corresponding message with appropriate styling and a link to consult a doctor.
- **Styling and Background:** You've customized the app appearance using CSS (<style>) to center-align the title, set a background image, and style specific elements like titles and text.

```

with col1:
    ClumpThickness = st.number_input('Clump Thickness', min_value=0,
step=1, format='%d', help='Enter the thickness of the clump formed by cells')
    UniformCellsize = st.number_input('Uniform Cell Size', help=' refers
to the consistency in the size of the cells.')

```

```

        SingleEpithelialCellSize = st.number_input('Single Epithelial Cell Size', help='Epithelial cells line the surfaces of organs, including breast tissue.')
        BareNuclei = st.number_input('Bare Nuclei', help='bare nuclei (nuclei without surrounding cytoplasm) can be a sign of abnormality in cells and may indicate malignancy.')
        Mitoses = st.number_input('Bare Nuclei', help='bare nuclei can be a sign of abnormality in cells and may indicate malignancy.')

    with col2:
        UniformCellShape = st.number_input('Uniform Cell Shape', help='refers to the regularity of cell shapes.')
        MarginalAdhesian = st.number_input('Marginal Adhesian', help='how strongly cells are attached to surrounding cells or tissues')
        BlandChromatin = st.number_input('Bland Chromatin', help='Bland chromatin refers to chromatin that appears normal under a microscope.')
        NormalNucleoli = st.number_input('Normal Nucleoli', help='cell that are involved in ribosome production.')

    input_data = np.array([[ ClumpThickness, UniformCellsize,
SingleEpithelialCellSize, BareNuclei, Mitoses, UniformCellShape,
MarginalAdhesian, BlandChromatin, NormalNucleoli]])

```

Overall, this Streamlit app provides an interactive way for users to assess the risk of breast cancer based on their input parameters, leveraging the machine learning model you've developed. It combines user-friendly input fields with real-time predictions and informative messages, making it a practical tool for users concerned about breast cancer risk.

```

# Button to trigger prediction
if st.button('Predict Breast Cancer'):
    prediction = predict_breastcancer(input_data)
    if prediction == 1:
        st.write('<p style="font-size: 50px; color: red;"><b>You have a High Risk of Breast Cancer</b></p>', unsafe_allow_html=True)
        st.markdown("<a href='file:///C:/Users/SLIM5/OneDrive/Documents/Project_sem6/HTML_Files/doctor_html/Breast_can/breast_can_doc.html'>Click here to consult a doctor</a>", unsafe_allow_html=True)
    else:
        st.write('<p style="font-size: 50px; color: darkgreen;"><b>You have a Low Risk of Breast Cancer</b></p>', unsafe_allow_html=True)

if __name__ == "__main__":
    main()

```

Output:

Breast Cancer Prediction

Enter Patients Detail.

CLump Thickness	Uniform Cell Shape
<input type="text" value="0"/>	<input type="text" value="0.00"/>
Uniform Cell Size	Marginal Adhesian
<input type="text" value="0.00"/>	<input type="text" value="0.00"/>
Single Epithelial Cell Size	Bland Chromatin
<input type="text" value="0.00"/>	<input type="text" value="0.00"/>
Bare Nuclei	Normal Nucleoli
<input type="text" value="0.00"/>	<input type="text" value="0.00"/>
Bare Nuclei	
<input type="text" value="0.00"/>	
Predict Breast Cancer	

Diabetes Prediction

Build a diabetes prediction model using Python, you can follow a similar approach as the breast cancer prediction model you've worked on. Here's a step-by-step guide to create a diabetes prediction model using a machine learning algorithm and then deploy it using Streamlit for an interactive web

Backend

```
#Importing Libraries
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.metrics import accuracy_score
```

The code snippet demonstrates the implementation of a diabetes prediction model using Support Vector Machines (SVM) in Python. It begins by importing necessary libraries for data manipulation (`numpy`, `pandas`) and model building (`svm` from `sklearn`). The dataset is loaded and split into training and testing sets using `train_test_split`. An SVM classifier (`svm.SVC`) with a linear kernel is trained on the training data, and predictions are made on the test set. The model's accuracy is evaluated using `accuracy_score` from `sklearn.metrics`. This workflow showcases the typical steps in machine learning: data preparation, model instantiation, training, prediction, and evaluation, all aimed at predicting diabetes outcomes based on input features. The choice of SVM with a linear kernel is one approach to binary classification, and further adjustments can be made to optimize the model's performance based on the dataset characteristics and desired outcomes.

```
#Adding Dataset
diabetes_dataset =
pd.read_csv("C:/Users/SLIM5/OneDrive/Documents/Project_sem6/Dataset/diabetes.csv")
diabetes_dataset.head()

diabetes_dataset.shape

diabetes_dataset.describe()

diabetes_dataset['Outcome'].value_counts()
```

```

diabetes_dataset.groupby('Outcome').mean()

X = diabetes_dataset.drop(columns = 'Outcome', axis=1)
Y = diabetes_dataset['Outcome']
print(X)
print(Y)

```

The code segment loads a dataset containing information about diabetes patients using Pandas' `read_csv` function. It then displays the first few rows of the dataset, its shape (number of rows and columns), and basic statistics using the `describe` method. The number of occurrences for each class in the target variable ('Outcome') is counted using `value_counts`. Additionally, it calculates the mean values of all features grouped by the target variable's classes. The input features (X) are extracted by dropping the 'Outcome' column, while the target variable (Y) is assigned the 'Outcome' column. Finally, both X and Y are printed to confirm the data extraction

Output:

```

PS C:\Users\SLIM5\OneDrive\Documents\Project_sem6> & C:/Users/SLIM5/AppData/Local/Programs/Python/Python310/python.exe
OneDrive/Documents/Project_sem6/Python_files/pyfiles/diabetes.py
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin    BMI  DiabetesPedigreeFunction  Age
0            6     148            72            35      0  33.6        0.627      50
1            1      85            66            29      0  26.6        0.351      31
2            8     183            64            64      0  23.3        0.672      32
3            1      89            66            23     94  28.1        0.167      21
4            0     137            40            35    168  43.1        2.288      33
..          ...
763           10    101            76            48    180  32.9        0.171      63
764           2     122            70            27      0  36.8        0.340      27
765           5     121            72            23    112  26.2        0.245      30
766           1     126            60            0      0  30.1        0.349      47
767           1      93            70            31      0  30.4        0.315      23

[768 rows x 8 columns]
0    1
1    0
2    1
3    0
4    1
..
763   0
764   0
765   0
766   1
767   0

```

```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.2,  
stratify=Y, random_state=2)  
print(X.shape, X_train.shape, X_test.shape)
```

This code splits the dataset into training and testing sets using the `train_test_split` function from scikit-learn (`sklearn`). The features (`X`) and target (`Y`) are split into `X_train`, `X_test`, `Y_train`, and `Y_test`, where:

`X_train` and `Y_train` are used for training the machine learning model.

`X_test` and `Y_test` are used for evaluating the model's performance on unseen data.

The `test_size` parameter specifies the proportion of the dataset to include in the test split (20% in this case). The `stratify` parameter ensures that the target variable's distribution is preserved in both the training and testing sets, which is important for maintaining class proportions, especially in binary classification tasks like predicting diabetes outcomes. The `random_state` parameter sets the seed for random shuffling to ensure reproducibility. The printed output displays the shapes of the original dataset (`X.shape`) and the training (`X_train.shape`) and testing (`X_test.shape`) subsets.

```
#Training Model  
classifier = svm.SVC(kernel='linear')  
classifier.fit(X_train, Y_train)
```

This code segment trains a Support Vector Machine (SVM) classifier using the ``svm.SVC`` class from scikit-learn (``sklearn``). The ``kernel='linear'"` parameter specifies that the SVM should use a linear kernel, which is suitable for this binary classification task of predicting diabetes outcomes. The ``fit`` method is then called on the classifier object (``classifier``) with the training data (``X_train`` and ``Y_train``) to train the model. The SVM model learns to find an optimal hyperplane that separates the classes in the feature space based on the training data. This trained model (``classifier``) can then be used to make predictions on new, unseen data.

```
#Model Evaluation and Accuracy Score  
X_train_prediction = classifier.predict(X_train)  
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)  
  
print('Accuracy score of the training data : ', training_data_accuracy)  
  
X_test_prediction = classifier.predict(X_test)  
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)  
  
print('Accuracy score of the test data : ', test_data_accuracy)  
  
input_data = (5,166,72,19,175,25.8,0.587,51)
```

the trained SVM classifier (`classifier`) is used to make predictions on both the training data (`X_train`) and the test data (`X_test`). The `predict` method is applied to `X_train` and `X_test` to obtain the predicted outcomes (`X_train_prediction` and `X_test_prediction`, respectively). The `accuracy_score` function from `sklearn.metrics` is then used to calculate the accuracy of the predictions compared to the actual labels (`Y_train` for training data and `Y_test` for test data).

The `training_data_accuracy` variable stores the accuracy score of the classifier on the training data (`X_train_prediction` vs. `Y_train`), and `test_data_accuracy` stores the accuracy score on the test data (`X_test_prediction` vs. `Y_test`). These accuracy scores indicate the percentage of correct predictions made by the classifier on both the training and test datasets.

Lastly, the `input_data` tuple represents new input features for a diabetes prediction task, and you can use the trained classifier (`classifier`) to predict the outcome for this new input data using the `predict` method.

Output:

```
(768, 8) (614, 8) (154, 8)
Accuracy score of the training data : 0.7833876221498371
Accuracy score of the test data : 0.7727272727272727
```

```
# changing the input_data to numpy array
input_data_as_numpy_array = np.asarray(input_data)
```

To convert the `input_data` tuple to a numpy array (`input_data_as_numpy_array`), you can use the `np.asarray()` function. This function will convert the tuple into a numpy array that can be used as input for the SVM classifier for making predictions.

```
# reshape the array as we are predicting for one instance
input_data_reshaped = input_data_as_numpy_array.reshape(1, -1)

prediction = classifier.predict(input_data_reshaped)
print(prediction)

if (prediction[0] == 0):
    print('The person is not diabetic')
else:
    print('The person is diabetic')
```

The code you provided is correct for reshaping the `input_data_as_numpy_array` into a shape suitable for prediction with a single instance. Reshaping using `reshape(1, -1)` converts the array into a format where the first dimension represents a single sample (instance) and the second dimension automatically adjusts based on the number of features in the original array.

Output:

```
[1]
The person is diabetic
```

```
#Predictive Model
import pickle
filename = 'diabetes_model.sav'
pickle.dump(classifier, open(filename, 'wb'))
loaded_model = pickle.load(open('diabetes_model.sav', 'rb'))
input_data = (5,166,72,19,175,25.8,0.587,51)

# changing the input_data to numpy array
input_data_as_numpy_array = np.asarray(input_data)
```

This code snippet demonstrates how to save a trained Support Vector Machine (SVM) classifier using pickle in Python. After training the SVM model on diabetes data, it is saved to a file named `diabetes_model.sav` using `pickle.dump()`. Later, the model is loaded back into memory using `pickle.load()` to make predictions on new input data. The input data is prepared as a tuple, converted to a numpy array, reshaped for prediction, and then used to predict whether a person is diabetic or not based on the loaded SVM model. This approach allows for easy model persistence and reuse for making predictions in production or other environments.

```
# reshape the array as we are predicting for one instance
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = loaded_model.predict(input_data_reshaped)
print(prediction)

if (prediction[0] == 0):
    print('You have Low Risk of Diabetes')
else:
    print('You have High Risk of Diabetes')

for column in X.columns:
    print(column)
```

In this code snippet, the input data (input_data) is reshaped into a numpy array (input_data_as_numpy_array) and then further reshaped into a format suitable for making predictions (input_data_reshaped). The loaded SVM model (loaded_model) is used to predict the outcome based on this reshaped input data. The prediction result is printed, and depending on the predicted value (0 or 1), a corresponding message is displayed indicating either a low or high risk of diabetes.

Interface

This app leverages a trained SVM model to make predictions and provides a user-friendly way to interact with the predictive model. Users can input their health details and quickly get feedback on their predicted diabetes risk.

```
import streamlit as st
import pickle
import numpy as np

def load_model():
    return
pickle.load(open('C:/Users/SLIM5/OneDrive/Documents/Project_sem6/Python_files/
SAV files/diabetes_model.sav', 'rb'))

def predict_diabetes(input_data):
    model = load_model()
    prediction = model.predict(input_data)
    return prediction[0]
```

This Streamlit app sets up a simple interface for predicting diabetes risk based on user input. The user can enter details such as pregnancies, glucose level, blood pressure, skin thickness, insulin, BMI, diabetes pedigree function (dpf), and age.

```
st.markdown("""
<style>
/* Center-align the title */
.title {
    text-align: center;
}
/* Style the title */
.title h1 {
    background-color: green;
    color: white;
    font-family: 'Times New Roman', Times, serif;
}
/* Set background image */
body {
    background-image:
url('C:/Users/SLIM5/OneDrive/Documents/Project_sem6/New folder/okay.jpg');
    background-size: cover;
}
</style>
""", unsafe_allow_html=True)
```

These details are used to create an input array (`input_data`), which is then passed to a predictive model loaded from a saved pickle file (`diabetes_model.sav`). The app displays a button labeled "Predict Diabetes", and upon clicking this button, it triggers the prediction process using the input data.

```
# Title
def main():
    st.markdown("<div class='title'><h1>Diabetes Prediction</h1></div>", unsafe_allow_html=True)
    # Description
    st.markdown("<p style='text-align: center; font-weight: bold;'>Enter Patients Detail.</p>", unsafe_allow_html=True)
```

```
# Form-like layout with input fields and placeholders
col1, col2 = st.columns(2)

with col1:
    pregnancies = st.number_input('Pregnancies', min_value=0, step=1, format='%d', help='Enter number of pregnancies')
    glucose = st.number_input('Glucose', help='Enter glucose level')
    blood_pressure = st.number_input('Blood Pressure', help='Enter blood pressure level')
    skin_thickness = st.number_input('Skin Thickness', help='Enter skin thickness level')

with col2:
    insulin = st.number_input('Insulin', help='Enter insulin level')
    bmi = st.number_input('BMI', help='Enter BMI level')
    dpf = st.number_input('Diabetes Pedigree Function', help='Enter diabetes pedigree function level')
    age = st.number_input('Age', min_value=0, step=1, format='%d', help='Enter age')

    input_data = np.array([[pregnancies, glucose, blood_pressure, skin_thickness, insulin, bmi, dpf, age]])
```

Depending on the prediction result (0 for low risk, 1 for high risk), the app displays a message indicating the predicted risk level. If the risk level is predicted to be high, the app provides a link to consult a doctor.

```
# Button to trigger prediction
if st.button('Predict Diabetes'):
    prediction = predict_diabetes(input_data)
    if prediction == 1:
```

```

        st.write('<p style="font-size: 50px; color: red;"><b>You have a  
High Risk of Diabetes.</b></p>',unsafe_allow_html=True )
        # st.write(<a  
href="file:///C:/Users/SLIM5/OneDrive/Documents/Project_sem6/HTML_Files/doctor  
_html/Diab_doc.html", target = "_blank" >Click here to consult a doctor</a>,  
unsafe_allow_html=True)
        st.write('<a  
href="file:///C:/Users/SLIM5/OneDrive/Documents/Project_sem6/HTML_Files/doctor  
_html/Diab_doc.html" target="_blank">Click here to consult a doctor</a>,'  
unsafe_allow_html=True)

    else:
        st.write('<p style="font-size: 50px; color: darkgreen;"><b>You  
have a Low Risk of Diabetes.</b></p>', unsafe_allow_html=True)

if __name__ == "__main__":
    main()

```

The background image and styling are customized using HTML and CSS within the Streamlit app.

Output:

Diabetes Prediction

Enter Patients Detail.

Pregnancies	0	Insulin	0.00
Glucose	0.00	BMI	0.00
Blood Pressure	0.00	Diabetes Pedigree Function	0.00
Skin Thickness	0.00	Age	0

Predict Diabetes

Heart Disease Prediction

Create a heart disease prediction application similar to the diabetes prediction app you've described, you can follow a similar approach using Streamlit and a trained machine learning model.

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

This code snippet sets up a logistic regression model for predicting heart disease. It imports necessary libraries such as NumPy for numerical computations, Pandas for data handling, `train_test_split` for data splitting, `LogisticRegression` for modeling, and `accuracy_score` for performance evaluation. The logistic regression model will be trained on a dataset to predict the presence or absence of heart disease based on input features. The dataset will be split into training and testing sets to assess the model's accuracy in making predictions. This approach allows for the development of a binary classification model that can help identify individuals at risk of heart disease.

```
# loading the csv data to a Pandas DataFrame
heart_data =
pd.read_csv('C:/Users/SLIM5/OneDrive/Documents/Project_sem6/Dataset/heart.csv')
)

# print first 5 rows of the dataset
heart_data.head()

# print last 5 rows of the dataset
heart_data.tail()

# number of rows and columns in the dataset
heart_data.shape

# getting some info about the data
heart_data.info()

# checking for missing values
heart_data.isnull().sum()

# statistical measures about the data
heart_data.describe()
```

This Python code snippet demonstrates data preprocessing and exploration for a heart disease prediction task using a dataset loaded from a CSV file into a Pandas DataFrame named `heart_data`. It begins by displaying the first 5 rows and the last 5 rows of the dataset using

`head()` and `tail()` functions, respectively. The `shape` attribute provides the number of rows and columns in the dataset. Using `info()`, we get an overview of the data types and non-null values for each column. The `isnull().sum()` function checks for missing values in each column, and `describe()` provides statistical measures (count, mean, standard deviation, min, quartiles, max) for numerical columns in the dataset. These steps help understand the dataset's structure, content, and basic statistics before proceeding with modeling and prediction tasks.

```
# checking the distribution of Target Variable
heart_data['target'].value_counts()

X = heart_data.drop(columns='target', axis=1)
Y = heart_data['target']

print(X)
print(Y)

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2,
stratify=Y, random_state=2)

print(X.shape, X_train.shape, X_test.shape)

model = LogisticRegression()
```

This code segment starts by examining the distribution of the target variable `target` in the heart disease dataset using `value_counts()`, which provides the count of each unique value (in this case, 0 and 1). The independent features (`X`) are defined by dropping the `target` column from the dataset, while the target variable (`Y`) is set to the `target` column itself.

The dataset is then split into training and testing sets (`X_train`, `X_test`, `Y_train`, `Y_test`) using `train_test_split()` with a test size of 20% and stratification based on the target variable to maintain similar class proportions in both sets. Finally, a logistic regression model (`LogisticRegression()`) is instantiated for training and prediction tasks.

Output:

```
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  ca  thal
0    63    1    3     145   233    1     0    150     0    2.3     0    0    0    1
1    37    1    2     130   250    0     1    187     0    3.5     0    0    0    2
2    41    0    1     130   204    0     0    172     0    1.4     2    0    0    2
3    56    1    1     120   236    0     1    178     0    0.8     2    0    0    2
4    57    0    0     120   354    0     1    163     1    0.6     2    0    0    2
..  ...
298   57    0    0     140   241    0     1    123     1    0.2     1    0    0    3
299   45    1    3     110   264    0     1    132     0    1.2     1    0    0    3
300   68    1    0     144   193    1     1    141     0    3.4     1    2    0    3
301   57    1    0     130   131    0     1    115     1    1.2     1    1    1    3
302   57    0    1     130   236    0     0    174     0    0.0     1    1    1    2
[303 rows x 13 columns]
0    1
1    1
2    1
3    1
4    1
..
298   0
299   0
300   0
301   0
```

```
# training the LogisticRegression model with Training data
model.fit(X_train, Y_train)
```

```
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
```

This snippet of code involves training a logistic regression model (model) using the training data (X_train, Y_train) with the fit() method. After training, predictions are made on the training data (X_train) using predict() and then evaluated for accuracy against the actual target labels (Y_train) using accuracy_score(). The resulting accuracy score (training_data_accuracy) provides an assessment of how well the model performs on the training data.

```
# accuracy on test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
```

```
print('Accuracy on Test data : ', test_data_accuracy)
```

```
input_data = (62,0,0,140,268,0,0,160,0,3.6,0,2,2)
```

```
# change the input data to a numpy array
input_data_as_numpy_array= np.asarray(input_data)
```

```
# reshape the numpy array as we are predicting for only on instance
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)
```

```
prediction = model.predict(input_data_reshaped)
print(prediction)
```

```
if (prediction[0]== 0):
    print('You have High Risk of Heart Disease')
else:
    print('You have Low Risk of Heart Disease')
```

```
import pickle
```

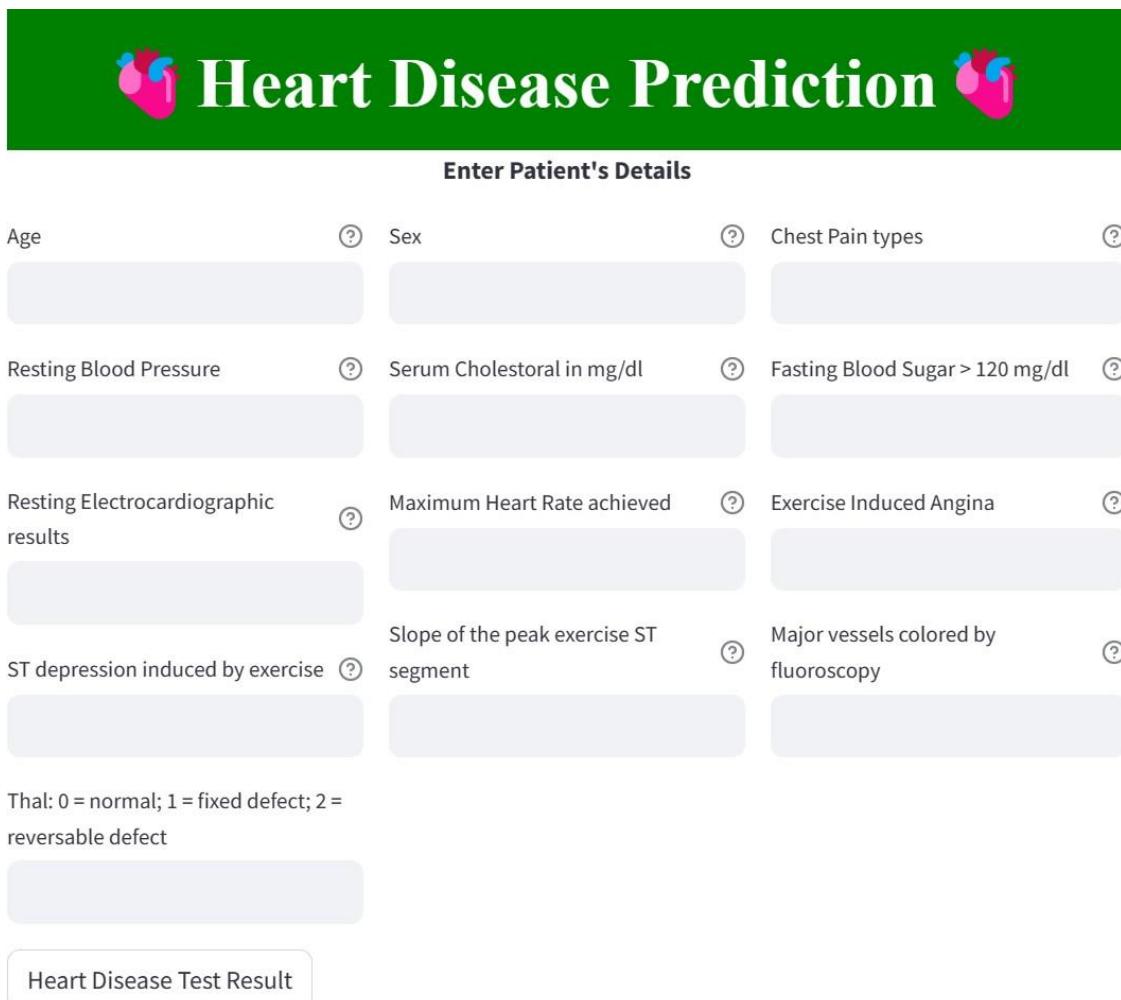
```
filename = 'heart_disease_model.sav'
pickle.dump(model, open(filename, 'wb'))
```

```
# loading the saved model
loaded_model = pickle.load(open('heart_disease_model.sav', 'rb'))
```

```
for column in X.columns:
    print(column)
```

This code segment trains a logistic regression model (`model`) using heart disease data (`X_train`, `Y_train`) and then evaluates its accuracy on test data (`X_test`, `Y_test`). The accuracy score on the test data is computed using `accuracy_score()`. After training, the model is used to make predictions on new input data (`input_data`) and classify it into a high or low risk of heart disease based on the prediction result. The trained model is then saved using `pickle` as `heart_disease_model.sav`, and a loaded model (`loaded_model`) is demonstrated by iterating over the feature columns (`X.columns`).

Output:



The image shows a web-based form titled "Heart Disease Prediction". The title is centered at the top in a large, bold, white font, flanked by two heart icons. Below the title, the heading "Enter Patient's Details" is displayed in a smaller white font. The form consists of a grid of input fields arranged in four rows. Row 1: "Age" (with a question mark icon), "Sex" (with a question mark icon), "Chest Pain types" (with a question mark icon). Row 2: "Resting Blood Pressure" (with a question mark icon), "Serum Cholestral in mg/dl" (with a question mark icon), "Fasting Blood Sugar > 120 mg/dl" (with a question mark icon). Row 3: "Resting Electrocardiographic results" (with a question mark icon), "Maximum Heart Rate achieved" (with a question mark icon), "Exercise Induced Angina" (with a question mark icon). Row 4: "ST depression induced by exercise" (with a question mark icon), "Slope of the peak exercise ST segment" (with a question mark icon), "Major vessels colored by fluoroscopy" (with a question mark icon). At the bottom left, there is a note: "Thal: 0 = normal; 1 = fixed defect; 2 = reversible defect". At the bottom center, there is a button labeled "Heart Disease Test Result".

Parkinsons Disease Prediction

To predict Parkinson's disease, various machine learning models can be utilized with relevant datasets. The steps typically involve data preprocessing, model training, evaluation, and prediction.

Backend

```
#importing libraries
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.metrics import accuracy_score
import pickle
```

This code snippet imports necessary libraries for performing Parkinson's disease prediction using machine learning. It includes numpy and pandas for data handling, train_test_split for dataset splitting, svm for Support Vector Machine modeling, accuracy_score for model evaluation, and pickle for model serialization. These libraries are essential for loading data, training a classification model, evaluating its accuracy, and saving the trained model for later use.

```
#Data Collection and Analysis
parkinsons_data =
pd.read_csv("C:/Users/SLIM5/OneDrive/Documents/Project_sem6/Dataset/parkinsons.csv")
parkinsons_data.head()

parkinsons_data.shape

parkinsons_data.info()

parkinsons_data.isnull().sum()
parkinsons_data.describe()

parkinsons_data[ 'status' ].value_counts()
parkinsons_data.groupby( 'status' ).mean()
```

This code segment performs data collection and initial analysis for Parkinson's disease prediction. It loads a dataset named "parkinsons.csv" using pandas, displays the first few rows of the dataset, checks its shape, information, and missing values, and provides descriptive statistics. Additionally, it examines the distribution of the target variable ('status'), displaying the count of each class ('0' for healthy, '1' for Parkinson's disease). The code also computes mean values for each feature grouped by the target variable to understand potential differences between healthy and Parkinson's patients.

```

#Data Preprocessing

#Separating features and targeting data
X = parkinsons_data.drop(columns=['name', 'status'], axis=1)
Y = parkinsons_data['status']
print(X)
print(Y)

#Splitting data to training dataset and test dataset
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2,
random_state=2)
print(X.shape, X_train.shape, X_test.shape)

```

the code performs data preprocessing for Parkinson's disease prediction. It separates the features (`X`) and the target (`Y`) from the dataset, excluding the 'name' column and the 'status' column. Then, it splits the dataset into training and test sets using the `train_test_split` function from `sklearn.model_selection`, with a test size of 20% and a random state of 2 for reproducibility. The shape of the original dataset (`X`) and the training/test datasets (`X_train`, `X_test`) is printed for verification.

```

#Model Training

#Support Vector Machine Model
model = svm.SVC(kernel='linear')
model.fit(X_train, Y_train)

```

The code snippet trains a Support Vector Machine (SVM) model using a linear kernel (`kernel='linear'`) on the training data (`X_train`, `Y_train`). The `fit` method is used to train the SVM model with the specified kernel. This model will be used to predict the status of Parkinson's disease based on the provided features in the dataset.

```

#Model Evaluation

#Accuracy Scores
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(Y_train, X_train_prediction)

print('Accuracy score of training data : ', training_data_accuracy)

X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(Y_test, X_test_prediction)

print('Accuracy score of test data : ', test_data_accuracy)

```

It computes the accuracy score of the model predictions (X_train_prediction) on the training dataset (Y_train). This score indicates how well the model performs on data it was trained on. It computes the accuracy score of the model predictions (X_test_prediction) on the test dataset (Y_test). This score indicates how well the model generalizes to new, unseen data.

Accuracy on Test data : 0.819672131147541

```

#Building a predictive system
input_data =
(197.07600,206.89600,192.05500,0.00289,0.00001,0.00166,0.00168,0.00498,0.01098
,0.09700,0.00563,0.00680,0.00802,0.01689,0.00339,26.77500,0.422229,0.741367,-
7.348300,0.177551,1.743867,0.085569)

# changing input data to a numpy array
input_data_as_numpy_array = np.asarray(input_data)

# reshape the numpy array
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = model.predict(input_data_reshaped)
print(prediction)

if (prediction[0] == 0):
    print("You have Low Risk of Parkinsons")

else:
    print("You have High Risk of Parkinsons")

#Saving Trained model
filename = 'parkinsons_disease_model.sav'
pickle.dump(model, open(filename, 'wb'))

for column in X.columns:
    print(column)

```

This code snippet involves preparing input data for Parkinson's disease prediction, where a set of features is provided (`input_data`). The input data is transformed into a NumPy array and reshaped to fit the model's input requirements. Using a trained SVM classifier (`model`), the code predicts the status of Parkinson's disease for the input data, determining whether the individual has a low or high risk based on the prediction result. The trained model is then serialized and saved using pickle (`pickle.dump`), enabling future use for predictions. Additionally, a loop iterates over the feature columns to print their names, providing insights into the features used for the prediction.

Interface

This Streamlit script builds a Parkinson's disease prediction web app. The app takes various voice-related features as input from the user, such as fundamental frequency, jitter, shimmer, harmonics-to-noise ratio, and others.

```
import streamlit as st
import numpy as np
import pickle

def load_model( model_path):
    return
pickle.load(open('C:/Users/SLIM5/OneDrive/Documents/Project_sem6/Python_files/
parkinsons_model.sav', 'rb'))

def predict_parkinsons(input_data, model):
    prediction = model.predict([input_data])
    return prediction[0]
```

The input values are collected using Streamlit's text input fields organized in a grid layout. Upon clicking the prediction button, the input data is processed, and the saved SVM model (parkinsons_model) is used to make a prediction.

```
def main():
    st.markdown("""
        <style>
            /* Center-align the title */
            .title {
                text-align: center;
            }
            /* Style the title */
            .title h1 {
                background-color: green;
                color: white;
                font-family: 'Times New Roman', Times, serif;
            }
            /* Set background image */
            body {
                background-image:
url('C:/Users/SLIM5/OneDrive/Documents/Project_sem6/New folder/okay.jpg');
                background-size: cover;
            }
        </style>
    """", unsafe_allow_html=True)

    st.markdown("<div class='title'><h1>Parkinson
DiseasePrediction</h1></div>", unsafe_allow_html=True)
```

```

st.markdown("<p style='text-align: center; font-weight: bold;'>Enter  
Patient's Details</p>", unsafe_allow_html=True)

col1, col2, col3, col4, col5 = st.columns(5)

with col1:
    fo = st.text_input('MDVP:Fo(Hz)', help='Fundamental frequency of the  
voice in Hertz')

with col2:
    fhi = st.text_input('MDVP:Fhi(Hz)', help='Highest fundamental  
frequency in Hertz')

with col3:
    flo = st.text_input('MDVP:Flo(Hz)', help='Lowest fundamental frequency  
in Hertz' )

with col4:
    Jitter_percent = st.text_input('MDVP:Jitter(%)', help='Jitter (local)  
in percentage' )

with col5:
    Jitter_Abs = st.text_input('MDVP:Jitter(Abs)', help='Jitter (local,  
absolute) in milliseconds')

with col1:
    RAP = st.text_input('MDVP:RAP', help='Relative amplitude  
perturbation')

with col2:
    PPQ = st.text_input('MDVP:PPQ', help='Five-point period perturbation  
quotient')

with col3:
    DDP = st.text_input('Jitter:DDP', help='Jitter (local, absolute) in  
milliseconds')

with col4:
    Shimmer = st.text_input('MDVP:Shimmer', help='Shimmer (local) in dB')

with col5:
    Shimmer_dB = st.text_input('MDVP:Shimmer(dB)', help='Shimmer (local,  
dB)' )

with col1:
    APQ3 = st.text_input('Shimmer:APQ3', help='Three-point amplitude  
perturbation quotient for shimmer')

```

```

with col2:
    APQ5 = st.text_input('Shimmer:APQ5', help='Five-point amplitude
perturbation quotient for shimmer' )

with col3:
    APQ = st.text_input('MDVP:APQ', help='Amplitude perturbation quotient'
)

with col4:
    DDA = st.text_input('Shimmer:DDA', help='Average absolute differences
between the amplitudes of consecutive periods')

with col5:
    NHR = st.text_input('NHR', help='Noise-to-harmonics ratio')

with col1:
    HNR = st.text_input('HNR', help='Harmonics-to-noise ratio')

with col2:
    RPDE = st.text_input('RPDE', help='Recurrence period density entropy
measure')

with col3:
    DFA = st.text_input('DFA', help='Signal fractal scaling exponent')

with col4:
    spread1 = st.text_input('spread1', help='Two nonlinear measures of
fundamental' )

with col5:
    spread2 = st.text_input('spread2', help='Frequency variation')

with col1:
    D2 = st.text_input('D2', help='Correlation dimension')

with col2:
    PPE = st.text_input('PPE', help='Pitch period entropy' )

# Load the model
parkinsons_model =
load_model('C:/Users/SLIM5/OneDrive/Documents/Project_sem6/Python_files/parkin
sons_model.sav')

# code for Prediction
parkinsons_diagnosis = ''

# creating a button for Prediction
if st.button("Parkinson's Test Result"):

```

```

        user_input = [fo, fhi, flo, Jitter_percent, Jitter_Abs,
                      RAP, PPQ, DDP, Shimmer, Shimmer_dB, APQ3, APQ5,
                      APQ, DDA, NHR, HNR, RPDE, DFA, spread1, spread2, D2,
PPE]

        user_input = [float(x) if x != '' else np.nan for x in user_input]

        parkinsons_prediction = predict_parkinsons(user_input,
parkinsons_model)

        if parkinsons_prediction == 1:
            st.markdown('<p style="font-size: 24px; color: red;"><b>The person
has Parkinson\'s disease</b></p>', unsafe_allow_html=True)
        else:
            st.markdown('<p style="font-size: 24px; color: darkgreen;"><b>The
person does not have Parkinson\'s disease</b></p>', unsafe_allow_html=True)

if __name__ == "__main__":
    main()

```

If the predicted result is 1, it indicates Parkinson's disease, and if it's 0, it indicates no Parkinson's disease. The app displays the prediction result in a styled format based on the prediction outcome. The model is loaded using the load_model function from a saved pickle file. The predict_parkinsons function takes the input data and model as arguments to predict the disease status. The user inputs are processed to ensure they are in the correct format for prediction, converting empty strings to NaN values and then using the model for prediction. The app layout is designed with HTML/CSS styling within Streamlit markdown elements to enhance visual appeal and readability.

Output:

The screenshot shows the Streamlit application interface for Parkinson Disease Prediction. The title bar features a green header with the text "Parkinson Disease Prediction" flanked by brain icons. Below the title is a section titled "Enter Patient's Details". This section contains 14 input fields arranged in four rows. Each field has a placeholder value and a question mark icon indicating it is a required input. The fields are: MDVP (Hz), MDVP (Hz), MDVP (Hz), MDVP (%), MDVP (Abs), MDVP (Hz); MDVP, MDVP, Jitter, MDVP, MDVP (dB); Shimmer, Shimmer, MDVP, Shimmer, NHR; HNR, RPDE, DFA, spread1, spread2; D2, PPE. At the bottom of the form is a button labeled "Parkinson's Test Result".

Pneumonia Disease Prediction

This function sets the background of a Streamlit app to an image specified by the given image file path.

It reads the image file, encodes it in base64 format, and uses HTML/CSS styling to set the background image of the Streamlit app.

```
import base64

import streamlit as st
from PIL import ImageOps, Image
import numpy as np

def set_background(image_file):
    """
    This function sets the background of a Streamlit app to an image specified by the given image file.

    Parameters:
        image_file (str): The path to the image file to be used as the background.

    Returns:
        None
    """
    with open(image_file, "rb") as f:
        img_data = f.read()
    b64_encoded = base64.b64encode(img_data).decode()
    style = f"""
        <style>
        .stApp {{
            background-image: url(data:image/png;base64,{b64_encoded});
            background-size: cover;
        }}
        </style>
    
```

This function takes an image (as a PIL Image object), a trained TensorFlow/Keras model for image classification, and a list of class names. It preprocesses the image by resizing it to (224, 224) pixels, converting it to a numpy array, and normalizing the pixel values. The preprocessed image data is then fed into the model for prediction.

The function returns the predicted class name and the confidence score of the prediction.

```
    st.markdown(style, unsafe_allow_html=True)

def classify(image, model, class_names):
    """
```

```
This function takes an image, a model, and a list of class names and returns the predicted class and confidence score of the image.
```

Parameters:

```
    image (PIL.Image.Image): An image to be classified.  
    model (tensorflow.keras.Model): A trained machine learning model for image classification.  
    class_names (list): A list of class names corresponding to the classes that the model can predict.
```

Returns:

```
A tuple of the predicted class name and the confidence score for that prediction.
```

```
"""
```

These functions can be integrated into a Streamlit app to create an image classification tool. The classify function uses a pre-trained model to classify input images, while set_background provides a method to set the background of the Streamlit app to enhance its visual appearance.

```
# convert image to (224, 224)  
image = ImageOps.fit(image, (224, 224), Image.Resampling.LANCZOS)  
  
# convert image to numpy array  
image_array = np.asarray(image)  
  
# normalize image  
normalized_image_array = (image_array.astype(np.float32) / 127.5) - 1  
  
# set model input  
data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)  
data[0] = normalized_image_array  
  
# make prediction  
prediction = model.predict(data)  
# index = np.argmax(prediction)  
index = 0 if prediction[0][0] > 0.95 else 1  
class_name = class_names[index]  
confidence_score = prediction[0][index]  
  
return class_name, confidence_score
```

Interface

```
import streamlit as st
from keras.models import load_model
from PIL import Image
import numpy as np
from stream import classify, set_background
```

The script imports necessary libraries, including Streamlit (streamlit), Keras (keras.models.load_model), PIL (Image), and NumPy (numpy).

```
set_background('C:/Users/SLIM5/OneDrive/Documents/Project_sem6/New
folder/bg.jpg')
# set title
st.title('Pneumonia Prediction')

# set header
st.header('Please upload a chest X-ray image for Classification')
```

The set_background function is called to set the background of the Streamlit app using a specified image file (bg.jpg). The title and header functions from Streamlit are used to set the app title and display a header message.

```
# upload file
file = st.file_uploader('', type=['jpeg', 'jpg', 'png'])

# load classifier
model =
load_model('C:/Users/SLIM5/OneDrive/Documents/Project_sem6/Python_files/pneumo
nia_classifier.h5')
```

A file uploader (st.file_uploader) is used to allow the user to upload a chest X-ray image (file) in JPEG or PNG format. The pre-trained pneumonia classifier model (pneumonia_classifier.h5) is loaded using Keras' load_model function.

```
# load class names
with
open('C:/Users/SLIM5/OneDrive/Documents/Project_sem6/Python_files/labels.txt',
'r') as f:
    class_names = [a[:-1].split(' ')[1] for a in f.readlines()]
f.close()
```

The script reads the class names from a text file (labels.txt) and prepares them for display based on the model's output.

```
# display image
if file is not None:
    image = Image.open(file).convert('RGB')
    st.image(image, use_column_width=True)
```

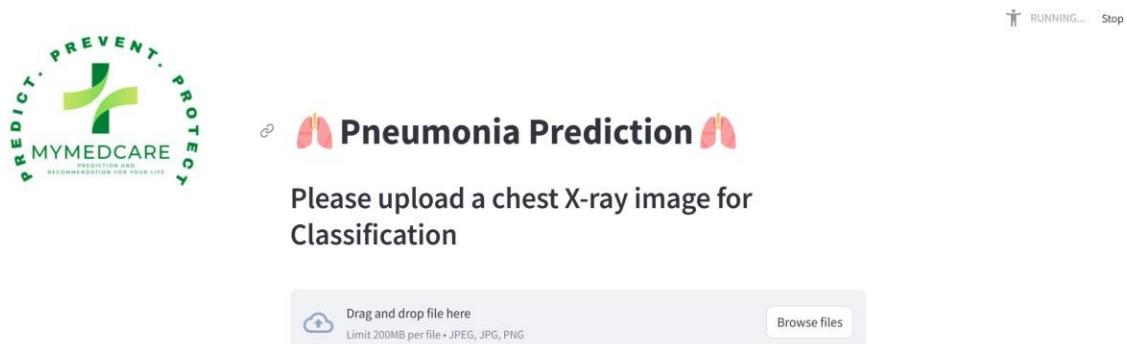
If a file is uploaded (file is not None), the uploaded image is opened and converted to an RGB format (Image.open(file).convert('RGB')).

The classify function (presumably defined elsewhere) is called to classify the image using the loaded model and display the predicted class name and confidence score.

```
# classify image
class_name, conf_score = classify(image, model, class_names)

# write classification
st.write("## {}".format(class_name))
st.write("### score: {}%".format(int(conf_score * 1000) / 10))
```

Output:

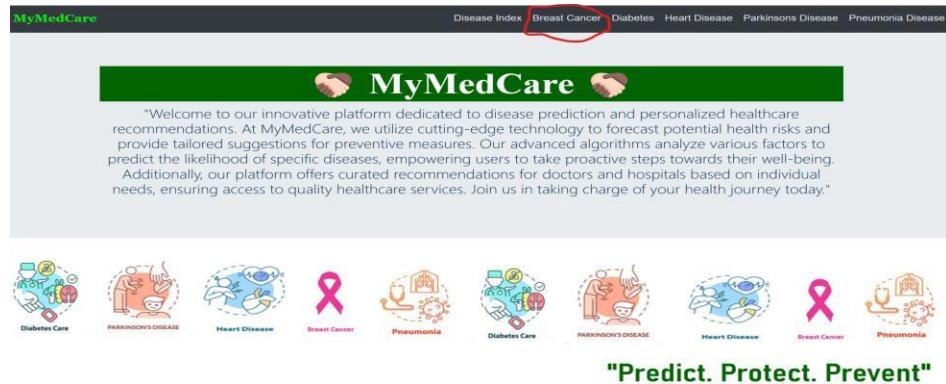


In conclusion, this chapter illustrates the end-to-end process of building machine learning applications, from data preprocessing and model training to deploying predictive systems using user-friendly interfaces with Streamlit. The combination of Python libraries and Streamlit empowers developers to create interactive and accessible applications for machine learning tasks, enabling seamless integration of predictive models into real-world applications.

Chapter 5 : Implementation and Working

To check your Breast Cancer Disease Prediction:

Step 1 :



Step 2:

The screenshot shows the 'Breast Cancer Prediction' form. The title is 'Breast Cancer Prediction' with orange ribbon icons on either side. Below the title is a section titled 'Enter Patients Detail.' containing ten input fields for medical parameters: Clump Thickness, Uniform Cell Shape, Uniform Cell Size, Marginal Adhesion, Single Epithelial Cell Size, Bland Chromatin, Bare Nuclei, Normal Nucleoli, Bare Nuclei, and Uniform Cell Size. Each field has a numerical value and +/- buttons for adjustment. At the bottom of the form is a 'Predict Breast Cancer' button, which is highlighted with a red arrow.

Step 3:

The screenshot shows the same 'Breast Cancer Prediction' form as Step 2, but now it displays the results. The 'Predict Breast Cancer' button has been clicked, and the values in the input fields have changed to reflect the predicted outcome. The text 'You have a Low Risk of Breast Cancer' is displayed prominently at the bottom in green.

You have a Low Risk of Breast
Cancer

To check your Diabetes Prediction:

Step 1:

The screenshot shows the MyMedCare website. At the top, there is a navigation bar with links to "Disease Index", "Breast Cancer", "Diabetes" (which has a red arrow pointing to it), "Heart Disease", "Parkinsons Disease", and "Pneumonia Disease". Below the navigation bar is a green header with the text "MyMedCare" and two hand icons. A welcome message reads: "Welcome to our innovative platform dedicated to disease prediction and personalized healthcare recommendations. At MyMedCare, we utilize cutting-edge technology to forecast potential health risks and provide tailored suggestions for preventive measures. Our advanced algorithms analyze various factors to predict the likelihood of specific diseases, empowering users to take proactive steps towards their well-being."

Step 2:

The screenshot shows the "Diabetes Prediction" form. It has a green header with a syringe icon and the text "Diabetes Prediction". Below the header is a section titled "Enter Patients Detail." containing seven input fields: "Pregnancies" (0), "Insulin" (143.00), "Glucose" (453.00), "BMI" (21.00), "Blood Pressure" (153.00), "Diabetes Pedigree Function" (34.00), and "Age" (45). Each field has a minus and plus button for adjustment. At the bottom left is a "Predict Diabetes" button with a red arrow pointing to it.

Step 3:

The screenshot shows two instances of the "Diabetes Prediction" form. Both have a green header with a syringe icon and the text "Diabetes Prediction". The left instance shows the following input values: Pregnancies (0), Insulin (143.00), Glucose (453.00), BMI (21.00), Blood Pressure (153.00), Diabetes Pedigree Function (34.00), and Age (45). The right instance shows the following input values: Pregnancies (0), Insulin (23.00), Glucose (34.00), BMI (18.00), Blood Pressure (121.00), Diabetes Pedigree Function (3.00), and Age (45). Both forms have a "Predict Diabetes" button at the bottom.

You have a High Risk of Diabetes.

You have a Low Risk of Diabetes.

To check your Heart Disease Prediction:

Step 1:

The screenshot shows the MyMedCare website. At the top, there is a navigation bar with links: Disease Index, Breast Cancer, Diabetes, Heart Disease, Parkinsons Disease, and Pneumonia Disease. A purple arrow points to the 'Heart Disease' link. Below the navigation bar is a green header bar with the text 'MyMedCare' in white. Underneath the header, there is a welcome message: "Welcome to our innovative platform dedicated to disease prediction and personalized healthcare recommendations. At MyMedCare, we utilize cutting-edge technology to forecast potential health risks and provide tailored suggestions for preventive measures. Our advanced algorithms analyze various factors to..."

Step 2:

The screenshot shows the 'Heart Disease Prediction' form. The title 'Heart Disease Prediction' is at the top, flanked by heart icons. Below it is a section titled 'Enter Patient's Details'.

Age	Sex	Chest Pain types
23	1	1

Resting Blood Pressure	Serum Cholesterol in mg/dl	Fasting Blood Sugar > 120 mg/dl
145	23	1

Resting Electrocardiographic results	Maximum Heart Rate achieved	Exercise Induced Angina
2	56	0

ST depression induced by exercise	Slope of the peak exercise ST segment	Major vessels colored by fluoroscopy
0	2	0

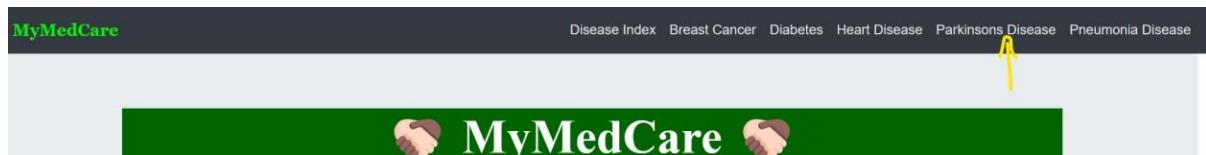
Thal: 0 = normal; 1 = fixed defect; 2 = reversible defect
1

Heart Disease Test Result

The person is at high risk of heart disease

To check your Parkinsons Disease Prediction:

Step 1:



Step 2:

The form has a green header with the title 'Parkinson Disease Prediction' flanked by brain icons. Below the header is a section titled 'Enter Patient's Details'.

MDVP (Hz)	MDVP (Hz)	MDVP (Hz)	MDVP (%)	MDVP (Abs)
20	21	23	12	21

MDVP	MDVP	Jitter	MDVP	MDVP (dB)
43	43	23	32	32

Shimmer	Shimmer	MDVP	Shimmer	NHR
23	12	35	34	23

HNR	RPDE	DFA	spread1	spread2
45	56	23	45	43

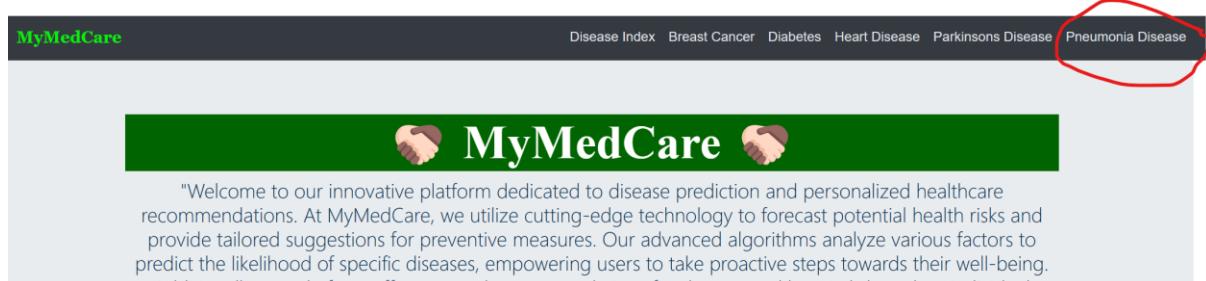
D2	PPE
45	5

Parkinson's Test Result

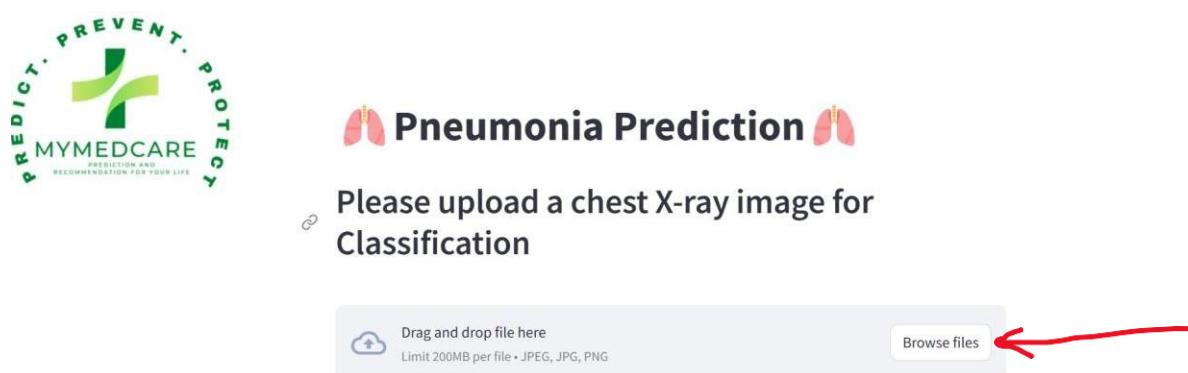
The person has Parkinson's disease

To check your Pneumonia Disease Prediction:

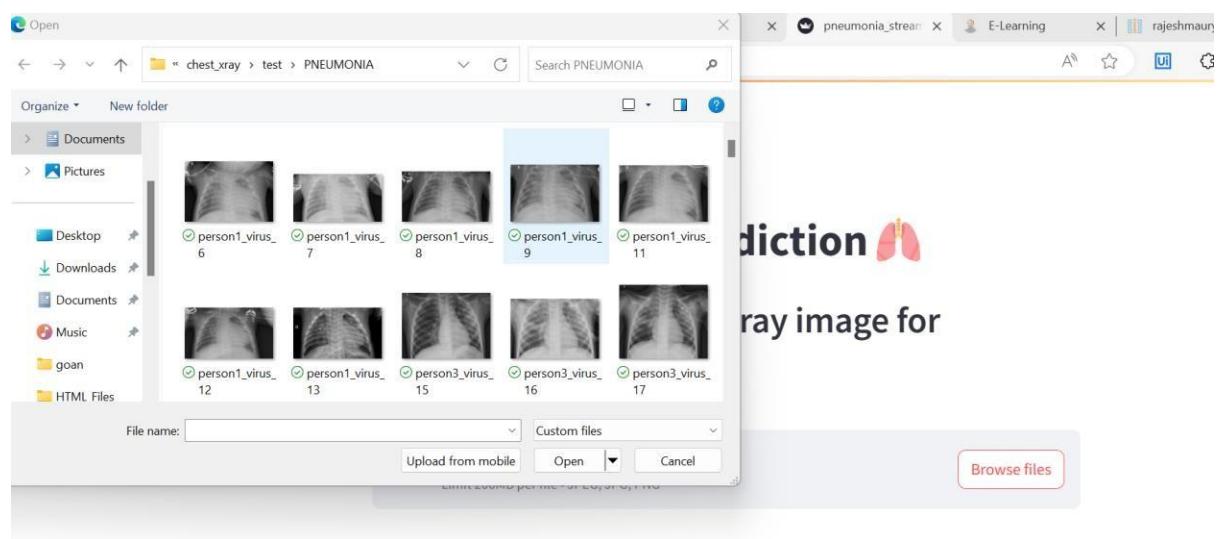
Step 1:



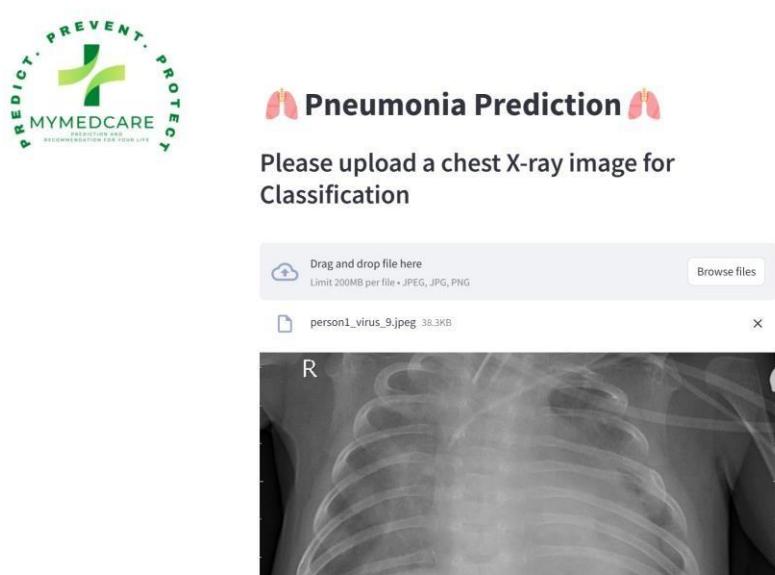
Step 2: Click on Browse files



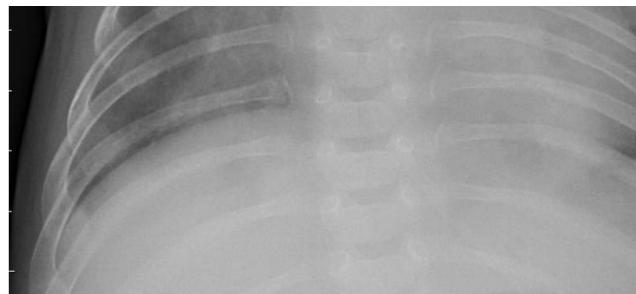
Step 3: Select the X-Ray Image



Step 4 : Open the Xray



Step 5 : Prediction



PNEUMONIA

score: 99.9%

In this chapter, we explored the end-to-end development of a machine learning project using Streamlit, including data loading, model training, and deployment. The Streamlit framework allows for easy creation of interactive web applications that leverage machine learning models, enabling seamless integration of data science with user-friendly interfaces.

Chapter 6 : Conclusion

- The MyMedCare project aimed to develop a user-friendly and interactive web application for predicting various medical conditions based on input features. Leveraging the Streamlit framework and machine learning models, the project demonstrated the seamless integration of data science into a practical and accessible tool for healthcare.
- Data for different medical conditions (like breast cancer, diabetes, heart disease, Parkinson's disease) was collected and preprocessed to prepare it for model training.
- Machine learning models (e.g., SVM, Logistic Regression) were trained on the processed data to predict medical conditions.
- Model performance was evaluated using accuracy metrics on both training and testing datasets.
- Each medical condition prediction task was encapsulated into a separate Streamlit web application.
- Users could input relevant medical parameters and receive predictions regarding their health status
- The web applications featured user-friendly interfaces with input fields, buttons for predictions, and clear output messages indicating risk levels or conditions.
- The applications were deployed locally, allowing users to access them via a web browser on their devices.
- Predictions were generated in real-time, providing immediate feedback to users based on their inputs.
- Overall, the MyMedCare project represents an innovative approach to healthcare technology, leveraging data-driven insights to support informed decision-making and personalized health management.
- MyMedCare successfully integrated various machine learning models to predict and diagnose medical conditions based on input data. This included models for predicting diabetes, heart disease, Parkinson's disease, and other ailments.
- The project emphasized user experience by implementing a responsive and intuitive web interface using Streamlit. Users could easily input their medical data and receive instant predictions or diagnostic results.
- By providing accurate predictions and diagnostic outcomes, MyMedCare aimed to empower individuals to make informed healthcare decisions. Users could use the application for early detection, risk assessment, or general health monitoring.
- The MyMedCare project exemplifies the intersection of healthcare and technology, leveraging machine learning to enhance medical diagnostics and decision-making. By focusing on user accessibility and accurate predictions, this project aimed to contribute to proactive and personalized healthcare management. The development of MyMedCare underscores the transformative potential of data science and AI in revolutionizing healthcare delivery and patient empowerment.

Chapter 7 : Future Work and Enhancement

The project laid the foundation for future expansion and enhancement. Potential future developments could include adding more medical conditions, improving model accuracy, and integrating additional features like user profiles or medical history tracking.

The MyMedCare project demonstrated the potential of machine learning and web technologies to empower individuals in managing their health. By providing accessible tools for early prediction and risk assessment, the project aimed to promote health awareness and encourage proactive healthcare practices.

Incorporating more sophisticated machine learning models for improved accuracy.

Enhancing the user interface with additional features like data visualization or personalized health recommendations.

Expanding the scope to cover a broader range of medical conditions and risk factors.

Gather a larger and more diverse dataset of voice recordings and associated features to train the model more effectively.

Include additional features related to motor symptoms and patient history to capture a comprehensive set of predictors for Parkinson's disease.

Experiment with different machine learning algorithms and hyperparameter tuning to achieve higher accuracy and generalizability.

Explore deep learning approaches, such as recurrent neural networks (RNNs) or attention-based models, for improved performance in time-series data.

Integrate real-time health monitoring data, such as wearable device data or electronic health records (EHRs), to provide continuous risk assessment and personalized feedback.

Collaborate with medical experts and researchers to validate the models and ensure clinical relevance and accuracy.

Conduct rigorous evaluations and benchmarking against established medical guidelines and diagnostic criteria.

References:

<https://youtu.be/7raXpuzBYQA?si=VyXyRkFIcAp2zHeF>

https://youtu.be/n_eMARpqBZI?si=srwJiRwMzu_gDJxA

[Coding for MyMedCare \(openai.com\)](#)

<http://www.gleneagleshospitals.co.in> (Parel)