LIVER DISEASE RISK PREDICTION BASED ON LIFESTYLE FACTORS USING BINARY CLASSIFICATION

SUBMITTED TO

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA

In the partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

MUMMADISETTI ANUSHA	**20NG1A05A3**
MANIKALA SUBRAHMANYAM	**20NG1A05A1**
ABDUL TAAHEER BAJI	**20NG1A0565**
YEGINATI BRAHMAIAH	**20NG1A05C6**

Under the Esteemed Guidance of

Dr. B V PRAVEEN KUMAR

Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

USHARAMA COLLEGE OF ENGINEERING AND TECHNOLOGY

AUTONOMOUS

(Approved by AICTE and JNTUK, Kakinada)

(ON NH 16, TELAPROLU, NEAR GANNAVARAM - 521109)

2020-2024

AUTONOMOUS

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(Affiliated of to JNTU Kakinada, Approved by A.I.C.T.E, New Delhi)

TELAPROLU, UNGUTURU MANDAL, KRISHNA DISTRICT-521109

2020-2024

CERTIFICATE

This is to certify that this project entitled **"LIVER DISEASE RISK PREDICTION BASED ON LIFESTYLE FACTORS USING BINARY CLASSIFICATION"** is the bonafide work of **M. Anusha (20NG1A05A3), M. Subrahmanyam (20NG1A05A1), A. Taaheer baji (20NG1A0565), Y. Brahmaiah (20NG1A05C6)** who carried out the work under my supervision, and submitted in partial fulfilment of the requirements for the award of the degree in Bachelor of Technology in Computer Science & Engineering, during the academic year 2020-24.

Project Guide

Dr. B V PRAVEEN KUMAR

Associate Professor

Head of the Department

Dr. K P N V SATYA SREE

Signature of External Examiner

DECLARATION

We hereby declare that the project entitled **"LIVER DISEASE RISK PREDICTION BASED ON LIFESTYLE ATTRIBUTES USING BINARY CLASSIFICATION"** is the work done by us during the academic year 2020-2024 and is submitted in partial fulfilment of the requirements for the award of degree of **Bachelor of technology** in **COMPUTER SCIENCE AND ENGINEERING** from **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA.**

BY

MUMMADISETTI ANUSHA	**20NG1A05A3**
MANIKALA SUBRAHMANYAM	**20NG1A05A1**
ABDUL TAAHEER BAJI	**20NG1A0565**
YEGINATI BRAHMAIAH	**20NG1A05C6**

ACKNOWLEDGEMENT

We express our sincere thanks where it is due

We are pleased to acknowledge our sincere thanks to our Honorable Chairman **SRI. S. RAMABRAHMAM GARU** for the support and advice which is given and for providing sufficient resources.

We are extremely thankful to **Dr. K RAJASEKHARA RAO**, Director of USHA RAMA COLLEGE OF ENGINEERING AND TECHNOLOGY, TELAPROLU for giving a golden opportunity to our education and project work.

We wish to avail this opportunity to express our thanks to **Dr. G V K S V PRASAD**, Principal, URCE for his continuous support and giving valuable suggestions during the entire period of the project work.

We take this opportunity to express our gratitude to **Dr. K P N V SATYA SREE**, Head of the Department Professor in **Computer Science and Engineering** for her valuable support and motivation at each and every point in successful completion of the project.

We Express our sincere thanks to our guide **Dr. B V PRAVEEN KUMAR, Associate Professor** in the Department of Computer Science and Engineering for motivating us to make our project successful. We grateful for his guidance and suggestions.

We also place our floral gratitude to all other teaching staff and lab technicians for their constant support and advice throughout the project.

BY

MUMMADISETTI ANUSHA	**20NG1A05A3**
MANIKALA SUBRAHMANYAM	**20NG1A05A1**
ABDUL TAAHEER BAJI	**20NG1A0565**
YEGINATI BRAHMAIAH	**20NG1A05C6**

LIVER DISEASE RISK PREDICTION BASED ON LIFESTYLE FACTORS USING BINARY CLASSIFICATION

ABSTRACT

ABSTRACT

Liver disease is a global health concern, and early detection is crucial for effective treatment. This research explores the application of machine learning algorithms to predict liver disease based on patient data. Binary classification is employed to distinguish patients with liver disease and without liver disease. This project aims to have a big impact on healthcare by helping to detect liver diseases early. Liver problems often show no symptoms until they become severe, so finding them early can save lives. We're using machine learning to make this process faster and more accurate.

This project aims to predict liver disease in patients based on various health attributes and demographic information. This research project focuses on the application of machine learning algorithms for the prediction of liver disease using lifestyle attributes as predictive features. The dataset utilized encompasses diverse lifestyle-related factors such as age, gender, alcohol intake, BMI, drug use, smoking status, and stress levels. In this project, we explore the application of machine learning for liver disease prediction. We start by preparing the data, performing feature selection, and evaluating three machine learning models: Logistic Regression, Support Vector Machine (SVM), and K-nearest Neighbors (KNN). The findings show the promise of machine learning in early liver disease detection, with the Support Vector Machine model achieving notable accuracy. These models have the potential to aid healthcare professionals and improve patient care.

Keywords— Liver Disease Prediction, Machine Learning, Data preprocessing, Data Cleaning, Feature Selection, Logistic Regression, Support Vector Machine (SVM), k-nearest Neighbors (KNN).

TABLE OF CONTENTS

CHAPTER - 1

INTRODUCTION

1. INTRODUCTION

Liver disease is a major global health concern, impacting millions of lives each year. Early diagnosis and effective intervention are pivotal in reducing the impact of liver disorders. In recent years, the integration of machine learning techniques into healthcare has shown remarkable promise in augmenting the early detection and management of various diseases, including liver conditions. This research project delves into the application of machine learning methodologies to enhance the prediction of liver disease, ultimately aiming to contribute to more timely and accurate diagnoses. The utilization of machine learning in the healthcare sector has seen exponential growth due to its ability to decipher patterns in complex medical data. Machine learning models, driven by large datasets and advanced algorithms, have exhibited remarkable capabilities in deciphering hidden trends, patterns, and relationships within medical data, thereby assisting healthcare professionals in making informed decisions. This research project adopts a systematic approach, beginning with data cleaning and data preprocessing to ensure data quality and consistency. It extends to feature selection, where the most relevant attributes are identified to improve the predictive accuracy of machine learning models. The selected features enable these models to focus on essential factors contributing to liver disease, thereby enhancing their diagnostic capabilities. To evaluate the effectiveness of machine learning in liver disease prediction, three distinct algorithms—Logistic Regression, Support Vector Machine (SVM), and K-nearest Neighbors (KNN)—are rigorously tested and compared. Each of these algorithms offers distinct strengths, rendering them suitable for early disease detection.

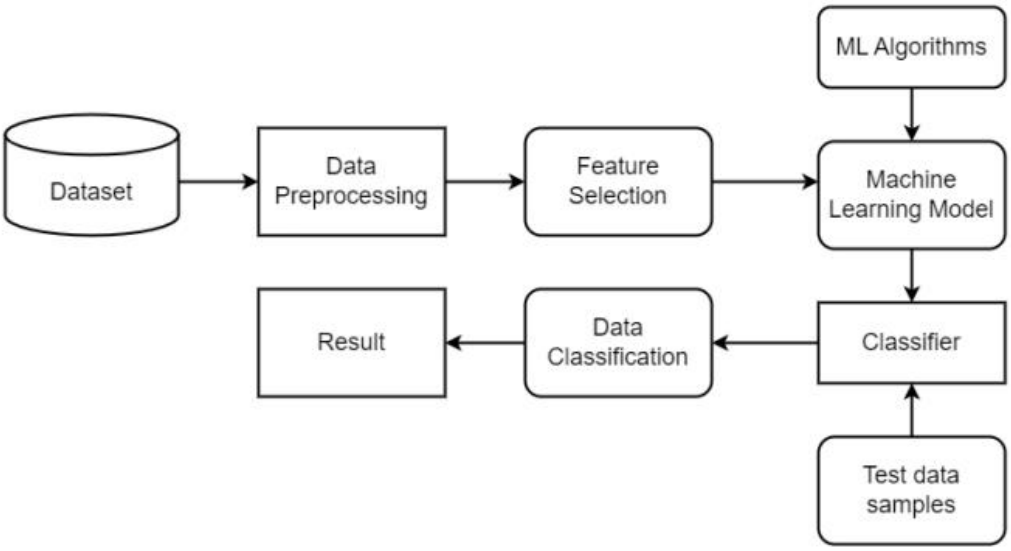


Fig:1 PROCESSING OF ML ALGORITHMS

1.1 LIVER DISEASE

Liver disease is a collective term that refers to a wide range of medical conditions and disorders that affect the liver, a vital organ in the human body. The liver plays a pivotal role in various essential physiological processes, including filtering the blood to remove toxins, metabolizing nutrients from the food we eat, and producing proteins necessary for clotting and other bodily functions. It also stores energy in the form of glycogen and helps to regulate various hormones. Liver Diseases can be caused by multiple factors, including viral infections (hepatitis), excessive alcohol consumption (alcoholic liver disease), obesity, drug-related damage (drug-induced liver injury), autoimmune responses (autoimmune hepatitis), genetic factors (genetic liver disorders), and the development of liver cancer. The effects of these diseases on the liver can range from mild inflammation to severe scarring (cirrhosis), which can ultimately lead to liver failure. Early-stage liver diseases may be asymptomatic or present with mild symptoms, such as fatigue or abdominal discomfort. However, as liver diseases progress, they can lead to more severe symptoms, including jaundice (yellowing of the skin and eyes), swelling in the abdomen, dark urine, and pale stools. In advanced cases, liver diseases can be life-threatening. Effective management of liver disease often requires early diagnosis, lifestyle modifications, medication, and in some cases, medical procedures or transplantation. The prevention and management of liver diseases are crucial for maintaining overall health and well-being, as the liver is indispensable for the proper functioning of the human body.

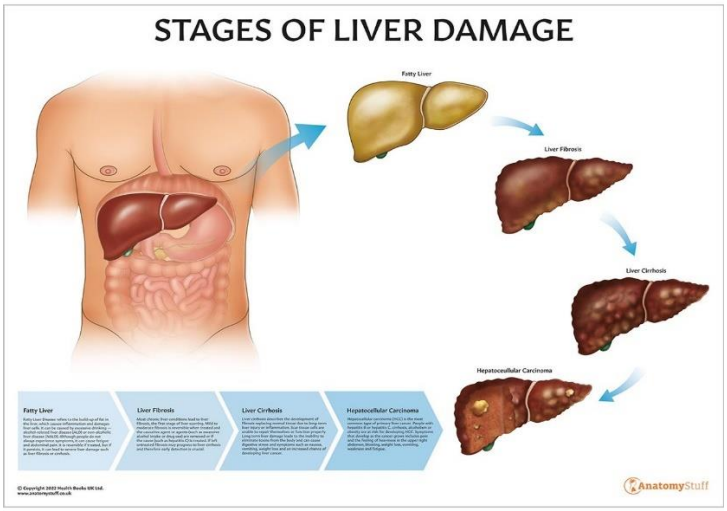


FIG:1.1 STAGES OF LIVER DAMAGE

STAGES OF LIVER DAMAGE

Inflammation: The initial stage of liver damage often involves inflammation of liver cells. This inflammation can be caused by factors such as viral infections (hepatitis B or C), excessive alcohol consumption, or autoimmune reactions. Inflammation may not always produce noticeable symptoms, but it is a sign of liver stress.

Fibrosis: As liver damage continues, inflammation can lead to the accumulation of fibrous tissue in the liver. This fibrosis represents the body's attempt to repair damaged liver cells. At this stage, the liver can still function adequately, and the damage may be reversible with appropriate medical intervention and lifestyle changes.

Cirrhosis: If liver damage persists and worsens, fibrosis can progress to cirrhosis. Cirrhosis is characterized by extensive scarring and distortion of the liver's structure. At this stage, liver function becomes severely compromised, and it may lead to significant health issues. Symptoms of cirrhosis can include jaundice, fluid retention in the abdomen (ascites), confusion, and easy bruising. Cirrhosis can be irreversible, but early diagnosis and intervention can help manage its progression.

Liver Cancer: In advanced cases of liver damage, particularly cirrhosis, there is an increased risk of developing liver cancer, known as hepatocellular carcinoma (HCC). Liver cancer can be life-threatening and may require treatments such as surgery, chemotherapy, or liver transplantation.

IMPACTS OF LIVER DISEASE

Liver disease can have a significant impact on an individual's health and well-being, leading to various physical and physiological consequences. The effects of liver disease can vary depending on the type and severity of the condition. Here are some common impacts of liver disease:

- **Jaundice**: One of the most noticeable signs of liver disease is jaundice, a yellowing of the skin and the whites of the eyes. Jaundice occurs when the liver is unable to effectively process bilirubin, a yellow pigment produced when red blood cells break down.
- **Fatigue**: Liver disease can lead to extreme fatigue and weakness, making it challenging to carry out daily activities and impacting a person's overall quality of life.
- **Abdominal Pain**: Many individuals with liver disease experience abdominal pain or discomfort, which can range from mild to severe. This pain may be due to liver inflammation or an enlarged liver.
- **Swelling**: Liver disease can cause fluid retention in the abdomen and legs, leading to swelling and discomfort. This condition is known as ascites.
- **Digestive Problems**: Liver disease can disrupt the production of bile, a substance essential for digesting fats. This can result in digestive issues, including diarrhea and nutrient malabsorption.
- **Dark Urine and Pale Stools**: Liver disease can alter the color and consistency of urine and stools. Dark urine and pale or clay-colored stools can be indicative of liver problems.
- **Bruising and Bleeding**: The liver produces clotting factors that help prevent excessive bleeding. Liver disease can lead to easy bruising and a higher risk of bleeding.
- **Cognitive Impairment**: Advanced liver disease, particularly cirrhosis, can lead to cognitive impairment and a condition known as hepatic encephalopathy. This can cause confusion, memory problems, and difficulty concentrating.
- **Increased Risk of Infections**: A compromised liver may weaken the immune system, making individuals more susceptible to infections.
- **Risk of Liver Cancer**: Certain forms of liver disease, such as cirrhosis, can increase the risk of developing liver cancer, which can be life-threatening.
- **Emotional and Psychological Impact**: Living with a chronic liver disease can take a toll on an individual's emotional and psychological well-being, leading to stress, anxiety, and depression.

It's important to note that the effects of liver disease can be managed and, in some cases, reversed with early diagnosis and appropriate medical care. Lifestyle modifications, including dietary changes and abstinence from alcohol, can also play a crucial role in mitigating the impact of liver disease. Regular medical check-ups and adherence to a healthcare provider's recommendations are vital for individuals with liver conditions.

LIFESTYLE FACTORS EFFECT THE LIVER DISEASE

Lifestyle factors play a crucial role in the development and progression of liver disease. Unhealthy lifestyle choices can significantly increase the risk of liver-related conditions. Here are some lifestyle factors that can impact liver health:

- **Excessive Alcohol Consumption:** Heavy and prolonged alcohol consumption is a leading cause of liver damage. It can lead to conditions like alcoholic liver disease, including alcoholic hepatitis and cirrhosis. Reducing or eliminating alcohol intake is essential for liver health.
- **Dietary Habits**: A diet high in saturated fats, sugars, and processed foods can contribute to obesity, insulin resistance, and non-alcoholic fatty liver disease (NAFLD). A balanced diet with plenty of fruits, vegetables, whole grains, and lean proteins can help prevent liver-related issues.
- **Physical Activity:** Sedentary lifestyles and lack of physical activity are associated with an increased risk of obesity and NAFLD. Regular exercise can help maintain a healthy weight and reduce the risk of liver disease.
- **Stress Levels**: Chronic stress can lead to unhealthy coping mechanisms, such as excessive alcohol consumption, unhealthy eating habits, and drug use. High stress levels can indirectly impact liver health. Stress management techniques, such as relaxation exercises and mindfulness, can help mitigate this risk.
- **Drug Use:** The misuse of prescription and illicit drugs, including opioids, can have a detrimental effect on the liver. Hepatitis C, a viral infection commonly associated with drug use, can also lead to liver disease.
- **Smoking:** Smoking is associated with an increased risk of liver cancer and can worsen the impact of other liver diseases.
- **Environmental Toxins:** Exposure to environmental toxins and pollutants, such as aflatoxins and industrial chemicals, can harm the liver. Occupational exposure to certain toxins is also a risk factor.
- **Obesity:** Being overweight or obese is a significant risk factor for liver disease, particularly NAFLD. Weight management through diet and exercise is essential in preventing liver-related issues.

Addressing these lifestyle factors and adopting healthy habits can significantly reduce the risk of liver disease and promote overall liver health. It's important to consult with healthcare professionals for guidance on lifestyle modifications, early detection, and the management of liver-related conditions.

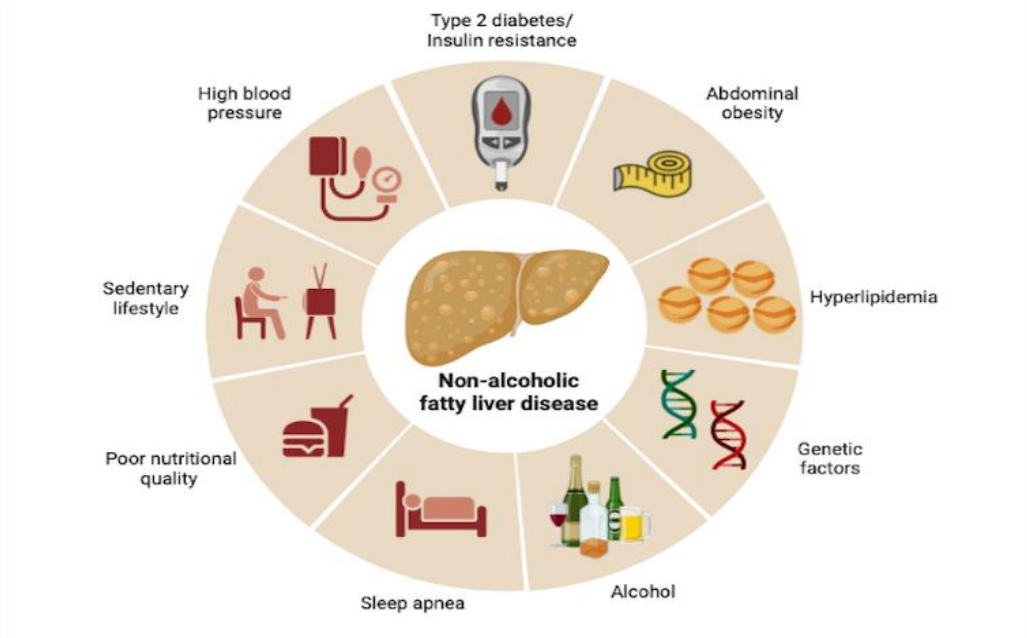


FIG:1.2 RISK FACTORS OF LIVER DISEASE

1.2 LITERATURE SURVEY

[1] The paper, "Software-based Liver Disease Prediction," emphasizes accessible healthcare by using software engineering to predict liver diseases. It employs various classification algorithms, such as Logistic Regression, SMO, Random Forest, Naive Bayes, J48, and knearest neighbor, to enhance prediction accuracy. The study also compares results with and without feature selection techniques. Its main contribution is the development of the ILDPS (Intelligent Liver Disease Prediction Software), combining feature selection and classification within a software engineering framework. The goal is to provide an innovative tool for liver disease prediction, improving healthcare accessibility and patient care.

[2] The paper, "Liver Disease Prediction Using Machine Learning Algorithms," addresses the significant global burden of liver diseases, which cause nearly a million deaths annually. The paper introduces the Liver Disease Prediction (LDP) method, aiming to serve healthcare professionals, students, and researchers. It assesses five machine learning algorithms: Support Vector Machine (SVM), Naïve Bayes, K-Nearest Neighbors (K-NN), Linear Discriminant Analysis (LDA), and Classification and Regression Trees (CART). Notably, K-NN achieves 91.7% accuracy, and an autoencoder network excels at 92.1%, promising early and accurate liver disease prediction.

[3] Neha Tanwar and Khandakar Faridar Rahman's paper, "Machine Learning in Liver Disease Diagnosis," from Banasthali Vidyapith, India, explores the application of machine learning in liver disease diagnosis. It underscores the increasing use of automatic decision-making systems in the medical field, driven by big data, deep learning, and machine learning concepts. These systems effectively extract insights from extensive medical datasets, aiding healthcare professionals in accurate and timely predictions and diagnoses of liver diseases. The paper acknowledges limitations in current studies and emphasizes the importance of future research to overcome these limitations and advance liver disease diagnosis using machine learning techniques.

[4] The paper titled "A Prediction Model of Detecting Liver Diseases in Patients using Logistic Regression of Machine Learning," presented at ICICC 2020, addresses the significant issue of liver diseases in India. With a high prevalence and late-stage diagnoses, it focuses on using Logistic Regression and machine learning to predict liver diseases early. This study underscores the potential of machine learning to aid healthcare in India.

[5] The paper "Detection of Liver Disease Using Machine Learning Approach" focuses on early diagnosis of liver diseases using machine learning. Liver diseases are a significant cause of global deaths, and their subtle symptoms often lead to late detection. The study uses advanced machine learning algorithms, including Support Vector Machine and K-Nearest Neighbors, to distinguish between individuals with and without liver disease. By leveraging AI and emphasizing preprocessing, feature extraction, and classification, the research seeks to improve diagnostic accuracy and proposes a hybrid classification system for better prediction. This work holds promise for healthcare professionals and researchers addressing the rising burden of liver disease worldwide

1.1.1 MACHINE LEARNING

A Machine Learning defined as "A computer program is said to learn from experience and from some tasks and some performance on, as measured by, improves with experience". Machine Learning is combination of correlations and relationships, most machine learning algorithms in existence are concerned with finding and/or exploiting relationship between datasets. Once Machine Learning Algorithms can pinpoint on certain correlations, the model can either use these relationships to predict future observations or generalize the data to reveal interesting patterns. In Machine Learning there are various types of algorithms such as Regression, Linear Regression, Logistic Regression, Naive Bayes Classifier, Bayes theorem, KNN (K-Nearest Neighbor Classifier), Decision Tress, Entropy, ID3, SVM (Support Vector Machines), K-means Algorithm, Random Forest and etc.,

The name machine learning was coined in 1959 by Arthur Samuel. Machine learning explores the study and construction of algorithms that can learn from and make predictions on data Machine learning is closely related to (and often overlaps with) computational statistics, which also focuses on prediction-making through the use of computers. It has strong ties to mathematical optimization, which delivers methods, theory and application domains to the field. Machine learning is sometimes conflated with datamining, where the latter subfield focuses more on exploratory data analysis and is known as unsupervised learning.

With in the field of data analytics, machine learning is a method used to devise complex models and algorithms that lend themselves to prediction; in commercial use, this is known as predictive analytics. These analytical models allow researchers, data scientists, engineers, and analysts to "produce reliable, repeatable decisions and results" and uncover "hidden in sights" through learning from historical relationships and trends in the data

Machine learning implementations are classified into three major categories, depending on the nature of the learning "signal" or "response" available to a learning system which are as follows:

Supervised learning

When an algorithm learns from example data and associated target responses that can consist of numeric values or string labels, such as classes or tags, in order to later predict the correct response when posed with new examples comes under the category of Supervised learning. This approach is indeed similar to human learning under the supervision of a teacher. The teacher provides good examples for the student to memorize, and the student then derives general rules from these specific examples.

Unsupervised learning

When an algorithm learns from plain examples without any associated response, leaving to the algorithm to determine the data patterns on its own. This type of algorithm tends to restructure the data into something else, such as new features that may represent a class or a new series of un-correlated values. They are quite useful in providing humans with insights into the meaning of data and new useful inputs to supervised machine learning algorithms. As a kind of learning, it resembles the methods humans use to figure out that certain objects or events are from the same class, such as by observing the degree of similarity between objects. Some recommendation systems that you find on the web in the form of marketing automation are based on this type of learning.

Reinforcement learning

When you present the algorithm with examples that lack labels, as in unsupervised learning. However, you can accompany an example with positive or negative feedback according to the solution the algorithm proposes comes under the category of Reinforcement learning, which is connected to applications for which the algorithm must make decisions (so the product is prescriptive, not just descriptive, as in unsupervised learning), and the decisions bear consequences. In the human world, it is just like learning by trial and error. Errors help you learn because they have a penalty added (cost, loss of time, regret, pain, and so on), teaching you that a certain course of action is less likely to succeed than others.

In this case, an application presents the algorithm with examples of specific situations, such as having the gamer stuck in a maze while avoiding an enemy. The application lets the algorithm know the outcome of actions it takes, and learning occurs while trying to avoid what it discovers to be dangerous and to pursue survival. You can have a look at how the company Google Deep Mind has created a reinforcement learning program that plays old Atari's video 3 games. When watching the video, notice how the program is initially clumsy and unskilled but steadily improves with training until it becomes a champion.

Semi-supervised learning

Where an incomplete training signal is given: a training set with some (often many) of the target outputs missing. There is a special case of this principle known as Transduction where the entire set of problem instances is known at learning time, except that part of the targets are missing. Supervised Learning the majority of practical machine learning uses supervised learning. Supervised learning is where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output.

$$Y = f(X)$$

The goal is to approximate the mapping function so well that when you have new input data(x)that you can predict the output variables (Y) for that data. It is called supervised learning because the process of an algorithm learning from the training dataset can be thought of as a teacher supervising the learning process. We know the correct answers, the algorithm iteratively makes predictions on the training data and is corrected by the teacher. Learning stops when the algorithm achieves an acceptable level of performance.

Types of Supervised Learning

Classification

It is a Supervised Learning task where output is having defined labels (discrete value). For example, in above Figure A, Output – Purchased has defined labels i.e., 0 or 1; 1 means the customer will purchase and 0 means that customer won't purchase. The goal here is to predict discrete values belonging to a particular class and evaluate on the basis of accuracy. It can be either binary or multi class classification. In binary classification, model predicts either 0 or 1; yes or no but in case of multi class classification, model predicts more than one class. Example: Gmail classifies mails in more than one classes like social, promotions, updates, forum.

Regression

It is a Supervised Learning task where output is having continuous value. Example in above Figure B, Output – Wind Speed is not having any discrete value but is continuous in the particular range. The goal here is to predict a value as much closer to actual output value as our model can and then evaluation is done by calculating error value. The smaller the error the greater the accuracy of our regression model.

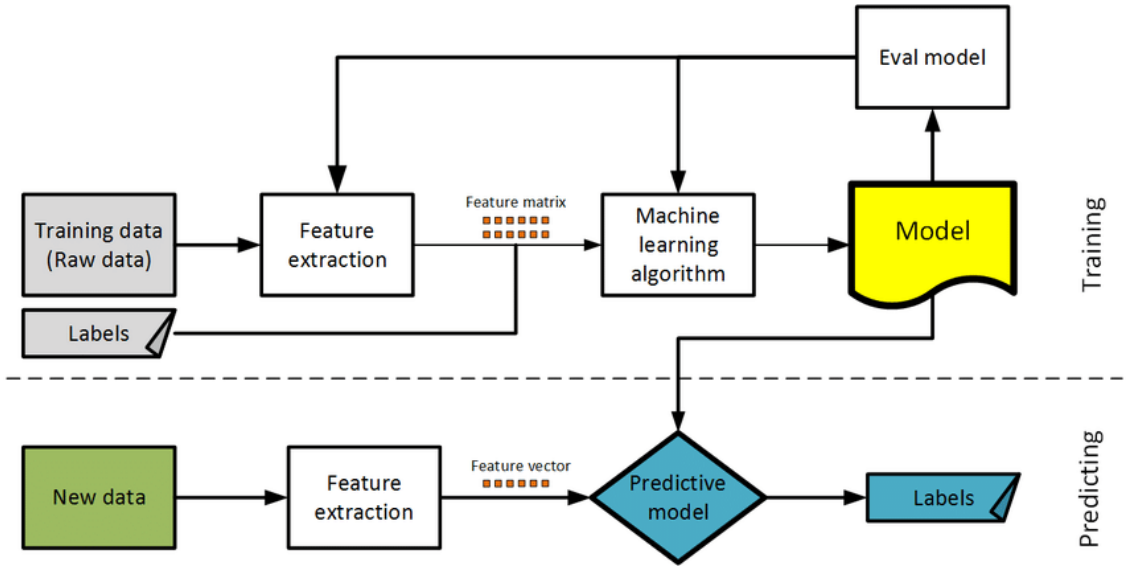


FIG: 1.1.1. FLOW CHART OF SUPERVISED LEARNING ALGORITHM

Classification

Data mining is the process of extracting knowledge-able information from huge amounts of data. It is an integration of multiple disciplines such as statistics, machine learning, neural networks and pattern recognition. Data mining extracts biomedical and health care knowledge for clinical decision making and generates scientific hypotheses from large medical data.

Association rule mining and classification are two major techniques of data mining. Association rule mining is an unsupervised learning method for discovering interesting patterns and their association in large data bases.

Classification is a supervised learning method used to find class labels for unknown samples. Classification is the task of assigning an object's tone of special predefined categories. It is pervasive problem that encompasses many applications.

Classification is designed as the task of learning a target function F that maps each attribute setA to one of the predefined class labels C. The target function is also known as classification model. A classification model is useful for mainly two purposes.

- Descriptive Modelling
- Predictive Modelling

Classification is the process of recognizing, understanding, and grouping ideas and objects into pre-set categories or "sub-populations." Using pre-categorized training datasets, machine learning programs use a variety of algorithms to classify future datasets into categories.

Classification algorithms in machine learning use input training data to predict the likelihood that subsequent data will fall into one of the predetermined categories. One of the most common uses of classification is filtering emails into "spam" or "non-spam."

In short, classification is a form of "pattern recognition," with classification algorithms applied to the training data to find the same pattern (similar words or sentiments, number sequences, etc.) in future sets of data.

Classification can be performed on structured or unstructured data. Classification is a technique where we categorize data into a given number of classes. The main goal of a classification problem is to identify the category/class to which a new data will fall under.

Few of the terminologies encountered in machine learning – classification:

- **Classifier:** An algorithm that maps the input data to a specific category.
- **Classification model:** A classification model tries to draw some conclusion from the input values given for training. It will predict the class labels/categories for the new data.
- **Feature:** A feature is an individual measurable property of a phenomenon being observed.
- **Binary Classification:** Classification task with two possible outcomes. E.g., Gender classification (Male / Female).
- **Multi-class classification:** Classification with more than two classes. In multi class classification each sample is assigned to one and only one target label. E.g., An animal can be cat or dog but not both at the same time.
- **Multi-label classification:** Classification task where each sample is mapped to a set of target labels (more than one class). E.g., A news article can be about sports, a person, and location at the same time.

Applications of Classification Algorithms:

- Email spam classification
- Bank customers loan pay willingness prediction.
- Cancer tumor cells identification.
- Sentiment analysis
- Drug's classification
- Facial key points detection
- Pedestrians' detection in an automotive car driving.

1.1.2 FEATURES OF MACHINE LEARNING

- It is nothing but automating the Automation.
- Getting computers to program themselves.
- Writing Software is bottleneck.
- Machine leaning models involves machines learning from data without the help of humans orany kind of human intervention.
- Machine Learning is the science of making of making the computers learn and act like humans by feeding data and information without being explicitly programmed.
- Machine Learning is totally different from traditionally programming, here data and output is given to the computer and in return it gives us the program which provides solution to the various problems. Below is the figure

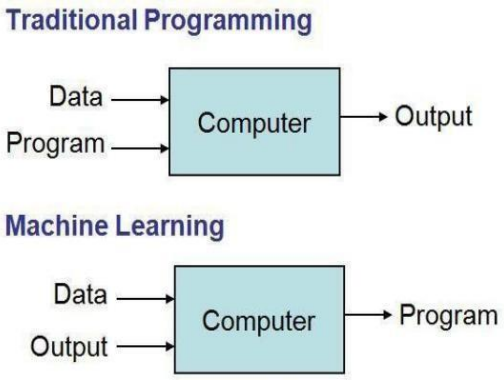
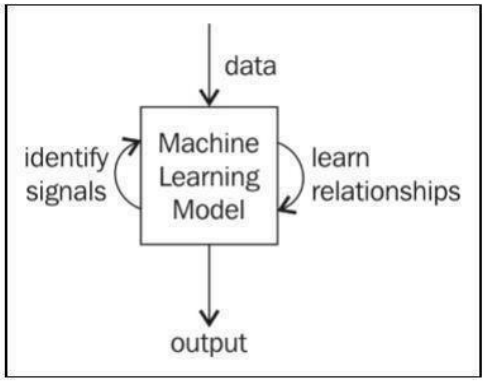


FIG: 1.1.1.1. TRADITIONAL PROGRAMMING VS MACHINE LEARNING

- Machine Learning is a combination of Algorithms, Datasets, and Programs.
- There here are Many Algorithms in Machine Learning through which we will provide us the exact solution in predicting the disease of the patients.
- How Does Machine Learning Works?
- Solution to the above question is Machine learning works by taking in data, findingrelationships within that data and then giving the output.



An overview of machine learning models

FIG: 1.1.1.2 MACHINE LEARNING MODEL

Confusion Matrix

A confusion matrix is a matrix that summarizes the performance of a machine learning model on a set of test data. It is often used to measure the performance of classification models, which aim to predict a categorical label for each input instance. The matrix displays the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) produced by the model on the test data.

For binary classification, the matrix will be of a 2X2 table, For multi-class classification, the matrix shape will be equal to the number of classes i.e for n classes it will be NXN

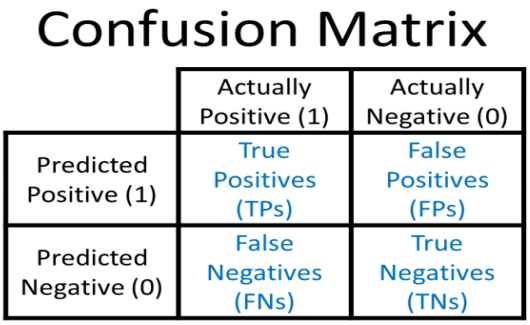


FIG: 1.1.1.3 CONFUSION MATRIX

TP here stands for True Positive predictions, for a binary classification problem like classifying the fraudulent transactions as 1, TP will give the count of the number of 1 s that were correctly classified as 1, i.e., number of fraudulent transactions that were classified as fraudulent. TN stands for true negative predictions, i.e., number of 0 s, non-fraudulent transactions, classified as 0. FP (False Positive) is the count of number of non-fraudulent transactions that were classified as fraudulent and FN (False Negative) is the count of number of fraudulent transactions that were classified as non-fraudulent.

There are various applications in which machine learning is implemented such as Web search, computing biology, finance, e-commerce, space exploration, robotics, social networks, debugging and much more.

SUPPORT VECTOR MACHINES

The support vector machine (SVM) is a machine learning and data mining algorithm to determine the strongest predictors of this variable for energy consumption. The research used popular classification methods to answer our question: best subset selection, boosting trees, and generalized additive models. Our first approach was to use forward, backward, and best subset selection to obtain a subset of predictors that most strongly predicted consumption with a linear relationship. The SVM provided an approach that was to use a tree-based method to stratify the predictor space into sample regions using recursive binary splitting. The research decided to use the boosting tree method, which is known to be one of the most potent tree-based models. SVM also has an excellent ability to deal with high dimensionality data.

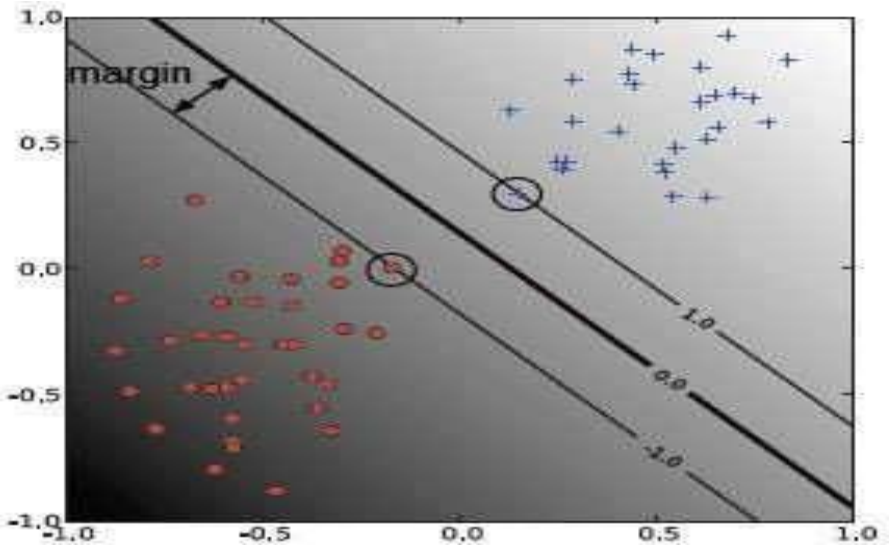


Fig: **1.1.1.2 SVM DIAGRAM**

LOGISTIC REGRESSION

Logistic regression, employed in the Supervised Learning technique, is one of the most widely used Machine Learning algorithms. It's a method for estimating a categorical dependent variable using a group of independent factors. Logistic regression is used to predict the contribution of a categorical dependent variable. As a result, the final value must be absolute or singleton. It might be yes or no, 0 or 1, true or false, and so on, but instead of even integers like 0 and 1, it returns probabilistic values in the middle. In terms of application, Logistic Regression is similar to Linear Regression. To solve regression issues, linear regression is employed, whereas logistic regression is utilized to solve classification issues.

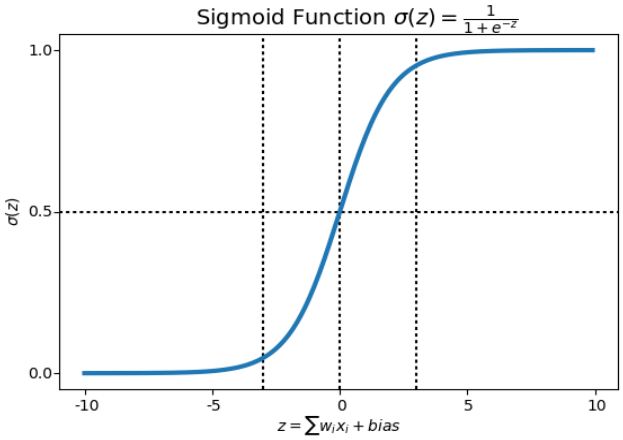


Fig: 1.1.1.3 LOGISTIC DIAGRAM

K-NEAREST NEIGHBOR

For both regression and classification, the supervised learning technique K-nearest neighbors (KNN) is utilized. KNN tries to predict the proper class for the test data by evaluating the difference between the test data and all training points. Then choose the K number of points that most closely resemble the test findings. The KNN method calculates the chance of test data belonging to any of the 'K' training data groups and then chooses the class with the highest probability. The average of the 'K' training points selected determines the significance in regression. Unlike the previous approaches, the k-nearest neighbor approach utilizes the data directly for classification rather than first developing a model.

Consequently, no additional model building is required, and the model's only variable is k, the number of closest neighbors to employ in estimating class membership: the value of p(y/x) is just the ratio of members of classy among the k nearest neighbors of x. Changing the value of k affects the model's stability (small or big values of k, respectively). The simplicity of usage of k-nearest neighbors over other algorithms is one of its main advantages. Neighbors can justify the categorization result; this case- based reasoning might be helpful in circumstances when black-box models are insufficient. The main drawback of k-nearest neighbors is that [53] Is the case neighborhood computation, which necessitates the definition of a metric that estimates the distance between data items. Assume we have two kinds, Category A and Category B, with a new data point x1. Would this data point fit into one of these categories? To address this kind of issue, a K-NN method is required. With the use of K-NN, we can rapidly categorize the type or class of a dataset.

The following algorithm can be used to illustrate how K-NN works:

- Initially the number of neighbors (K) is determined.
- Hence the Euclidean distance between the neighbors of K is determined.
- Then the nearest neighbors K is searched by using the measured Euclidean distance.
- Then count the number of data points for each group among these k- neighbors.
- The most recent data points are then assigned to the group that has the most neighbors.

finished the model

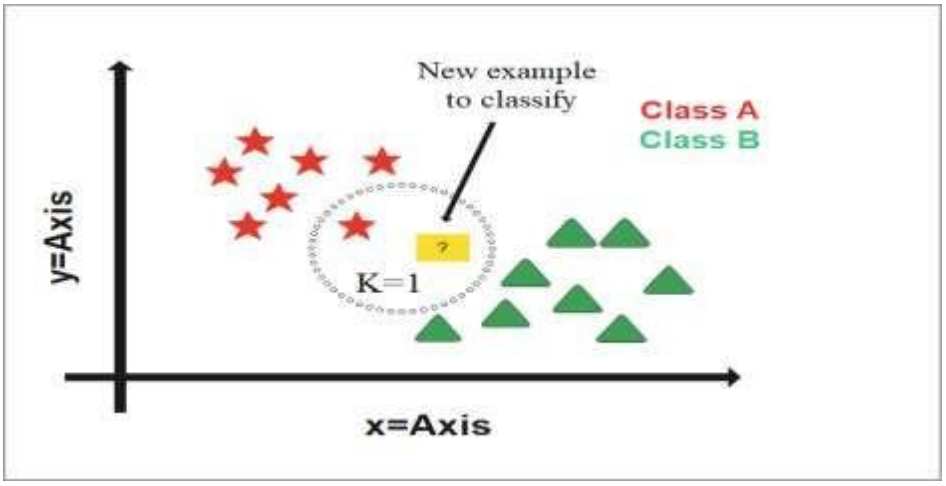


Fig: **1.1.1.4 KNN DIAGRAM**

There are no pre-defined mathematical methods for determining the most advantageous K value to choose the best K. Set a random K value as the starting point and begin computation. As K is set to a low value, the judgment boundaries become unpredictable. The higher the K value, the smoother the judgment boundaries get, which is better for classification.

1.1.3. EXISTING SYSTEM

The existing system for liver disease prediction hinges on a series of clinical tests that provide insights into an individual's liver health. These tests encompass a range of vital clinical parameters, including liver enzyme levels, bilirubin concentrations, albumin levels, and platelet counts. Liver enzymes like ALT and AST serve as markers for potential liver damage or inflammation. Bilirubin, a waste product, can indicate liver dysfunction when its levels are abnormal, often leading to jaundice. Albumin levels reflect the liver's synthetic capacity and may be indicative of liver disease when they deviate from the norm. Additionally, platelet counts, essential for blood clotting, are monitored to gauge the severity of liver disease.

To enhance the diagnostic process, machine learning algorithms are employed within the existing system. These algorithms play a pivotal role in analyzing the clinical data derived from the tests, effectively identifying patterns and relationships within the data. This analytical approach aids in making predictions about the presence or risk of liver disease, contributing to early detection and diagnosis.

However, this system is not without its limitations. It predominantly focuses on specific biomarkers and an individual's medical history, potentially sidelining lifestyle factors that can exert a substantial influence on liver health. Aspects such as diet, physical activity, stress levels, alcohol consumption, and smoking habits are not comprehensively considered within this framework. As a result, there exists a risk of missing diagnoses and incomplete risk assessments, as the system may not capture the entirety of factors impacting liver health. This exclusive reliance on clinical data may restrict the system's ability to provide a holistic evaluation of an individual's health and their susceptibility to liver disease.

Disadvantages of the Existing System

- Limited Scope: The existing system primarily relies on clinical data, such as blood tests and specific biomarkers. This limited scope may result in missed diagnoses and a failure to consider the broader health context of individuals.
- Lack of Lifestyle Factors: It does not incorporate lifestyle attributes, including diet, physical activity, stress levels, alcohol consumption, and smoking habits, which are known to impact liver health. This omission overlooks crucial risk factors.
- Incomplete Risk Assessment: The system's exclusive reliance on clinical data may lead to incomplete risk assessments, as it does not consider the holistic factors influencing an individual's liver health.

1.1.4. PROPOSED SYSTEM

In the proposed system, we aim to revolutionize the approach to liver disease prediction by shifting the focus from clinical data to lifestyle attributes. We acknowledge the substantial role that lifestyle factors play in liver health and their potential impact on early disease detection. Our system will incorporate data related to various lifestyle attributes, including diet, physical activity, stress levels, alcohol consumption, and smoking habits, among others. By doing so, we will create a more holistic and comprehensive model for liver disease prediction. This approach not only considers clinical data but also takes into account an individual's habits and choices, which can significantly impact liver health. By applying advanced machine learning algorithms to this new dataset, we intend to improve the accuracy and timeliness of liver disease prediction. Our proposed system offers the potential to identify individuals at risk of liver disease based on their lifestyles, enabling early interventions and personalized healthcare strategies. This innovative approach aligns with the growing importance of preventive medicine and individualized healthcare, ultimately leading to better health outcomes and a reduced burden of liver diseases on individuals and healthcare systems.

Advantages of the Proposed System

- Comprehensive Data: The proposed system addresses the limitations of the existing system by incorporating a wide array of lifestyle attributes. This comprehensive dataset provides a more holistic view of an individual's health.
- Early Detection: By considering lifestyle factors, the proposed system can enable early detection of liver disease based on an individual's habits and choices, potentially leading to timely interventions.
- Personalized Healthcare: The system's ability to identify individuals at risk of liver disease based on their lifestyles allows for personalized healthcare strategies, which can lead to better health outcomes and reduced healthcare costs.
- Preventive Medicine: The proposed system aligns with the growing importance of preventive medicine, marking a shift towards proactive healthcare strategies that can reduce the burden of liver diseases.
- Improved Accuracy: By incorporating lifestyle attributes and applying advanced machine learning algorithms, the proposed system aims to improve the accuracy of liver disease prediction, reducing the chances of missed diagnoses.

CHAPTER - 2

AIM&SCOPE

2. AIM & SCOPE

The primary aim of this project is to revolutionize the landscape of liver disease prediction by shifting the focus from clinical data to lifestyle attributes. The core objective is to address the limitations of the existing system, which predominantly relies on clinical data derived from blood tests and specific biomarkers. By doing so, we aspire to create a more comprehensive and holistic model for predicting liver disease.

Our project aims to bridge the gap between clinical data and lifestyle factors, recognizing the profound impact of the latter on liver health. Through the integration of data related to diet, physical activity, stress levels, alcohol consumption, smoking habits, and other lifestyle attributes, we aim to provide a more accurate and timely assessment of an individual's liver disease risk.

Furthermore, our project seeks to leverage advanced machine learning algorithms to analyze this enriched dataset effectively. By doing so, we aim to improve the accuracy of liver disease prediction and enable early detection. This proactive approach can lead to timely interventions and personalized healthcare strategies, thereby reducing the burden of liver diseases on individuals and healthcare systems.

Ultimately, the overarching goal of this project aligns with the growing importance of preventive medicine and individualized healthcare. We aspire to contribute to better health outcomes by empowering individuals to take charge of their liver health and providing healthcare professionals with a more comprehensive understanding of their patients' well-being. Through this innovative approach, we envision a future where liver diseases can be detected and managed more effectively, resulting in healthier lives for all.

2.1. FEASIBILITY STUDY:

The feasibility of the project is analyzed in this phase. During system analysis the feasibility study of the proposed system is to be carried out. For feasibility analysis, some understanding of the major requirements for the system is essential.

The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation.

A feasibility study for your project, "Liver Disease Risk Prediction Based on Lifestyle Attributes Using Binary Classification," is essential to assess the viability, practicality, and potential success of the project. The following sections present a detailed analysis of the feasibility of this project:

- Technical Feasibility
- Economic Feasibility
- Operational Feasibility
- Schedule Feasibility
- Legal and Ethical Feasibility
- Risk Assessment

2.1.1. Technical Feasibility

- Data Availability: The technical feasibility of the project relies on the availability of comprehensive and reliable datasets containing lifestyle attributes and corresponding liver disease outcomes. It is crucial to ensure that such data is accessible and of sufficient quality.
- Machine Learning Tools: The technical feasibility also depends on the availability of suitable machine learning tools and libraries. It is important to assess whether the chosen algorithms and technologies can effectively process and analyze the data.

2.1.2. Economic Feasibility

- Costs: Consider the financial feasibility of the project. This includes costs associated with data acquisition, hardware, software, human resources, and any other relevant expenses. Ensure that the project fits within the available budget or funding sources.
- Return on Investment (ROI): Evaluate the potential ROI, which could be indirect, such as improved healthcare outcomes and reduced healthcare costs due to early detection. The project's economic viability may also include the potential for commercialization if applicable.

2.1.3. Operational Feasibility

- Data Collection: Assess the practicality of collecting lifestyle attribute data from a representative population. Consider the logistics of data collection and potential challenges.
- Data Processing: Ensure that data preprocessing and feature selection methods can be efficiently implemented to prepare the data for machine learning.
- Scalability: Examine whether the system can handle varying data loads and adapt to different settings or healthcare facilities.

2.1.4. Schedule Feasibility

- Timeline: Develop a detailed project timeline that considers all phases, from data collection and preprocessing to model development and testing. Ensure that the project can be completed within the stipulated timeframe.

2.1.5. Legal and Ethical Feasibility

- Data Privacy: Consider the legal and ethical aspects related to data privacy and compliance with relevant regulations (e.g., GDPR or HIPAA). Ensure that all data usage adheres to ethical guidelines.
- Informed Consent: If the project involves human subjects, ensure that the necessary ethical approvals and informed consent procedures are in place.

2.1.6. Risk Assessment

- Identify potential risks that may affect the project's feasibility. This could include data quality issues, unexpected technical challenges, or ethical concerns. Develop mitigation strategies for these risks.

In summary, a comprehensive feasibility study is essential to determine whether your project is technically, economically, and operationally viable. It should also assess legal and ethical considerations, potential risks, and the support of stakeholders. By conducting this study, you can make informed decisions about the project's scope and ensure that it aligns with your objectives and resources.

2.2 SYSTEM REQUIREMENT SPECIFICATION

A Software Requirements Specification (SRS) – a requirements specification for software system– is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

System requirements specification is a structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions.

2.2.1. FUNCTIONAL REQUIREMENTS:

A Functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all cases where the system uses the functional requirements are captured in use cases. Functional requirements are supported by non-functional requirements (also known as quality requirements), which impose constraints on the design or implementation (such as performance requirements, security, or reliability).

As defined in requirements engineering, functional requirements specify particular results of a system. This should be contrasted with non-functional requirements which specify overall characteristics such as co stand reliability. Functional requirements drive the application architecture of a system, while non-functional requirements drive the technical architecture of a system.

- Functional Requirements concerns with the specific functions delivered by the system.
- So, functional requirements are statements of the services that the system must provide.
- The functional requirements of the system should be both complete and consistent
- Completeness means that all the services required by the user should be defined.
- Consistency means that requirements should not have any contradictory definitions.

The requirements are usually described in a fairly abstract way. However, functional system requirements describe the system function in details, its inputs and outputs, exceptions and soon.Take user id and password match it with corresponding file entries. If a match is found then continue else raise an error message.

2.2.2.NON-FUNCTIONAL REQUIREMENTS:

Non-functional Requirements refer to the constraints or restrictions on the system. They may relate to emergent system properties such as reliability, response time and store occupancy or the selection of language, platform, implementation techniques and tools.
The non-functional requirements can be built on the basis of needs of the user, budget constraints, organization policies and etc.

- **Performance Requirement:** Your project aims to accurately predict emotions from facial expressions in real-time video. Ensuring that the system performs with high accuracy and minimal latency aligns with this requirement.
- **Platform Constraints:** You mentioned building an intelligent system for predicting emotions. The choice of platforms, such as computer or mobile devices, may influence the system's accessibility and usability.
- **Accuracy and Precision:** These requirements are directly relevant to your project, as you need accurate and precise emotion predictions from the AI model.
- **Modifiability:** As your project evolves, making changes to improve the system's performance and accuracy will be essential. This relates to the effort required to modify the software.
- **Portability:** Your project's deployment on mobile phones showcases its portability and adaptability to different devices.
- **Reliability:** You need the facial emotion detection system to be reliable, as inaccuracies can lead to misinterpretations of emotions. You must specify the consequences of failure and strategies for prediction and correction.
- **Security:** Protecting user data and the system itself is crucial, especially if your project handles sensitive information.
- **Usability:** You should consider how user-friendly your application is. Users should find it easy to learn and operate, as reflected in metrics such as learning time.

ACCESSIBILITY:

Accessibility is a general term used to describe the degree to which a product, device, service, or environment is accessible by as many people as possible. In our project people who have registered with the registration page can access their data with the help of login. User interface is simple and efficient and easy to use

MAINTAINABILITY:

In software engineering, maintainability is the ease with which a software product can be modified in order to include new functionalities can be added in the project based on the user requirements just by adding the appropriate files to existing project using .net and programming languages. Since the programming is very simple, it is easier to find and correct the defects and to make the changes in the project.

SCALABILITY:

System is capable of handling increase total throughput under an increased load when resources (typically hardware) are added. System can work normally under situations such as low bandwidth and large number of users.

PORTABILITY:

Portability is one of the key concepts of high-level programming. Portability is the software code base feature to be able to reuse the existing code instead of creating new code when moving software from an environment to another. Project can be executed under different operation conditions provided it meet its minimum configurations. Only system files and dependent assemblies would have to be configured in such case.

VALIDATION:

It is the process of checking that a software system meets specifications and that it fulfills its intended purpose. It may also be referred to as software quality control. It is normally the responsibility of software testers as part of the software development life cycle. Software validation checks that the software product satisfies or fits the intended use (high-level checking), i.e., the software meets the user requirements, not as specification artifacts or as needs of those who will operate the software only but as the needs of all the stakeholders.

2.2.3. HARDWARE REQUIREMENTS

- System Processor : Intel i3 and above
- Hard Disk : 40 GB
- RAM : 4GB(Min)

2.2.4. SOFTWARE REQUIREMENTS

- Operating System : Windows
- Front-end : HTML, CSS, Java Script
- Back-end : MONGO DB
- Software Tools : Python (3.11.4) or Anaconda
- Packages : Pandas,Flask,Numpy,Sklearn,cv2

CHAPTER - 3

CONCEPTS & METHODS

3. CONCEPTS & METHODS

3.1. PROBLEM DEFINITION

The problem addressed by our project, "Liver Disease Risk Prediction Based on Lifestyle Attributes Using Binary Classification," centers around the limitations of traditional clinical data-based approaches to liver disease prediction. The existing system primarily relies on clinical data, including blood tests and specific biomarkers, to assess liver health and diagnose liver diseases. While this approach has been valuable in detecting liver issues, it has significant drawbacks.

One of the key problems lies in the narrow focus of clinical data. These tests primarily concentrate on specific liver enzymes, bilirubin, albumin levels, and platelet counts, offering a limited perspective on an individual's overall health. Clinical data may not encompass the full spectrum of factors influencing liver health, including lifestyle choices and habits that can significantly impact disease risk.

Furthermore, the existing system often lacks the ability to consider the holistic impact of lifestyle attributes on liver health. Lifestyle factors such as diet, physical activity, stress levels, alcohol consumption, and smoking habits are known to be pivotal in determining an individual's risk of liver disease. Neglecting these aspects can lead to missed diagnoses and incomplete risk assessments, limiting the effectiveness of disease prediction.

Therefore, the problem addressed by this project is twofold: the limited scope of clinical data in assessing liver health and the neglect of lifestyle attributes in existing prediction models. The goal is to create a more comprehensive and holistic approach to liver disease prediction, which leverages lifestyle attributes to provide a more accurate and timely assessment of an individual's liver disease risk. In doing so, we aim to bridge the gap between clinical data and lifestyle factors, ultimately enhancing the effectiveness of liver disease prediction and contributing to more proactive and personalized healthcare strategies.

3.2. PROJECT DESCRIPTION

The problem at the heart of this project stems from the traditional clinical data-based approach to liver disease prediction, which has inherent limitations. Existing systems predominantly rely on clinical data, such as blood tests measuring specific biomarkers, to evaluate liver health and diagnose liver diseases. While these methods have contributed significantly to the detection of liver issues, they are constrained by their narrow focus.

A central issue is the confined scope of clinical data. These tests primarily assess specific liver enzyme levels, bilirubin concentration, albumin levels, and platelet counts, providing a limited perspective on an individual's overall health. Clinical data frequently fails to capture the entire spectrum of factors that can influence liver health, omitting critical considerations such as lifestyle choices and habits that play a significant role in determining the risk of liver disease.

Furthermore, the current system often neglects the comprehensive impact of lifestyle attributes on liver health. Lifestyle factors, including dietary habits, physical activity, stress levels, alcohol consumption, and smoking practices, have been recognized as pivotal determinants of an individual's susceptibility to liver disease. The failure to incorporate these aspects in prediction models can lead to incomplete risk assessments, potentially resulting in missed diagnoses and undermining the predictive accuracy of the system.

Consequently, the problem in question encompasses two key aspects: the constrained perspective of clinical data in the assessment of liver health and the omission of lifestyle attributes in existing prediction models. The project's primary objective is to establish a more holistic and comprehensive approach to liver disease prediction, one that leverages lifestyle attributes to offer a more precise and timely evaluation of an individual's liver disease risk. By bridging the gap between clinical data and lifestyle factors, the project aims to enhance the effectiveness of liver disease prediction, contributing to more proactive, informed, and personalized healthcare strategies.

3.2.1. ALGORITHMS PROPOSED

Logistic Regression

Logistic Regression is a widely used machine learning algorithm for binary classification tasks, making it highly relevant for your project focused on liver disease prediction. This algorithm estimates the probability of an instance belonging to a specific class. It leverages the logistic function (also known as the sigmoid function) to transform the output into a probability value within the range of 0 to 1. The logistic regression formula calculates the odds of the instance being in one class compared to the other, and the sigmoid function maps this value to a probability. It's particularly useful when you want to understand the impact of different independent variables on the probability of a binary outcome, which, in your case, could be the likelihood of an individual having liver disease based on their lifestyle attributes. The algorithm optimizes the coefficients in such a way that it maximizes the likelihood of the observed data. It's a valuable tool for understanding the influence of lifestyle factors on liver disease risk.

Formula: In logistic regression, we use the logistic function (also called the sigmoid function) to model the probability of a binary outcome. The logistic function is defined as:

$$P(Y = 1) = 1 \div 1 + e^{-(b_0+b_1X_1+b_2X_2.+\cdots+b_pX_p)}$$

Where:

- P(Y=1) represents the probability that the dependent variable Y equals 1.
- e is the base of the natural logarithm.
- b0 is the intercept term, and b1, b2, ……, bp are the coefficients associated with the Independent variables X1, X2, ……, Xp.
- X1, X2, ……, Xp are the independent variables or features.

The logistic function maps any input to a value between 0 and 1, representing probabilities. If P(Y=1) is greater than or equal to 0.5, the model predicts the binary outcome as 1; otherwise, it predicts it as 0.

Support Vector Machine (SVM)

Support Vector Machine (SVM) is another robust algorithm suited for binary classification problems like liver disease prediction. SVM finds an optimal hyperplane that best separates data points into different classes. For your project, this means finding the best way to differentiate individuals who are at risk of liver disease from those who are not. What makes SVM powerful is its ability to find the maximum-margin hyperplane, the one that provides the largest separation between the two classes. This margin, essentially a gap between data points of different classes, allows for effective classification. SVM also identifies support vectors, which are data points closest to the decision boundary, ensuring a robust and accurate model. It's an ideal choice for handling complex datasets and can adapt well to nonlinear relationships between lifestyle factors and liver disease risk.

Formula: The core idea behind SVM is to find the optimal hyperplane that best separates data points of different classes. For a binary classification problem, the equation of this hyperplane can be represented as:

$$\mathbf{f(x)=sign\,(w\cdot x+ b)}$$

- f(x) is the classification function that predicts the class label of the data point.
- w is the weight vector that defines the orientation of the hyperplane.
- x represents the input features of the data point.
- b is the bias term or intercept.

K-Nearest Neighbors (KNN)

K-nearest Neighbors (KNN) is a versatile machine learning algorithm that operates on the principle of similarity. In the context of your project, KNN assesses the similarity between individuals based on their lifestyle attributes and classifies them accordingly. KNN is intuitive and straightforward to understand. When predicting liver disease risk, KNN calculates the distance between data points, with Euclidean distance being a common choice, and identifies the k-nearest neighbors of each data point. The class with the majority of the nearest neighbors determines the classification of that data point. KNN is non-parametric, meaning it doesn't make strong assumptions about the underlying data distribution. This flexibility makes KNN suitable for a variety of scenarios, especially when the relationship between lifestyle attributes and liver disease risk may not be linear or easily modeled by other techniques.

Formula (Euclidean Distance): The Euclidean distance between two points (x1, x2) and (y1, y2) in a two dimensional space is calculated as:

$$\mathbf{d = \sqrt{[(x2 - x1)2 + (y2 - y1)2]}}$$

3.2.2. METHODOLOGY

3.2.2.1. Data collection

In our research, we have compiled a dataset that specifically centers around the lifestyle attributes of individuals who could potentially be susceptible to liver disease. This dataset encompasses a wide range of information related to the daily habits, behaviors, and choices of these individuals, which might have a bearing on their liver health. Such lifestyle attributes could include dietary patterns, alcohol consumption, physical activity levels, smoking habits, and more. By meticulously collecting and analyzing data on these aspects, we aim to gain a deeper understanding of how various lifestyle factors may contribute to the development or prevention of liver disease. This dataset serves as a valuable resource for our research, enabling us to draw meaningful insights and potentially make recommendations for healthier lifestyles or early interventions to reduce the risk of liver disease.

Attribute	**Description**	**Data Type**
Age	Age of the individual	Numeric
Gender	Gender of the individual (Male, Female)	Categorical
Alcohol Intake	Level of alcohol intake	Numeric
Physical Activity	Level of physical activity	Numeric
BMI (Body Mass Index)	Body Mass Index	Numeric
Sleep Hours	Average number of hours of sleep per night	Numeric
Drug Use	Drug use status (No drug use, Drug use)	Categorical
Smoking Status	Smoking behaviour (Non-smoker, Smoker)	Categorical
Stress Levels	stress levels (Low, Moderate, High)	Numeric
Dataset	Dataset or group identifier (0 or 1)	Categorical

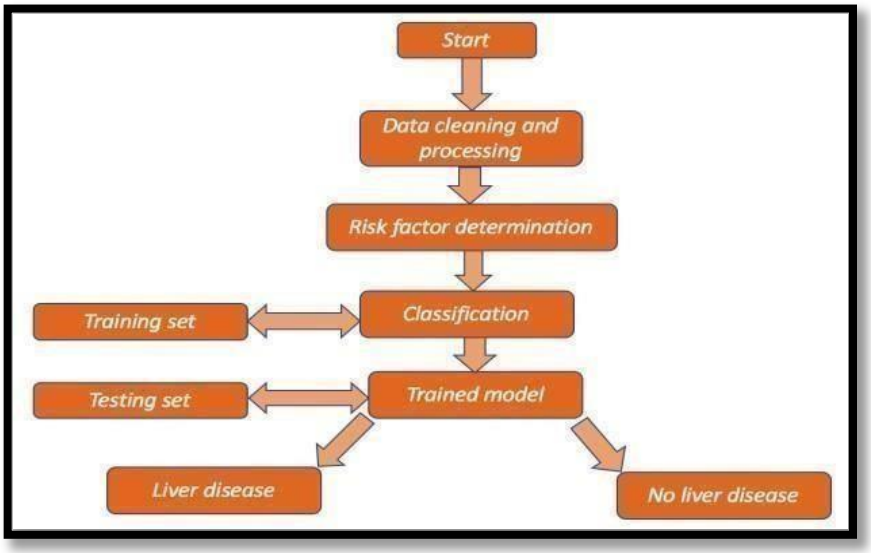
TABLE 3.2.2.1: THE PATIENT LIFESTYLE ATTRIBUTES FOR LIVER DISEASE

This table presents various lifestyle attributes of patients relevant to liver disease. These attributes include age, gender, alcohol intake, physical activity level, BMI (Body Mass Index), sleep hours, drug use status, smoking behavior, perceived stress levels, and a dataset identifier. The data types for these attributes vary, including numeric and categorical values.

3.2.2.2 Data Preprocessing and Data Cleaning

Data preprocessing is a crucial phase in preparing the dataset for the binary classification task of liver disease prediction. The following steps are involved:

1. Duplicate Handling: Duplicate entries, representing identical records, were systematically identified and removed from the dataset. This process eliminates data redundancy and ensures each patient is represented only once.



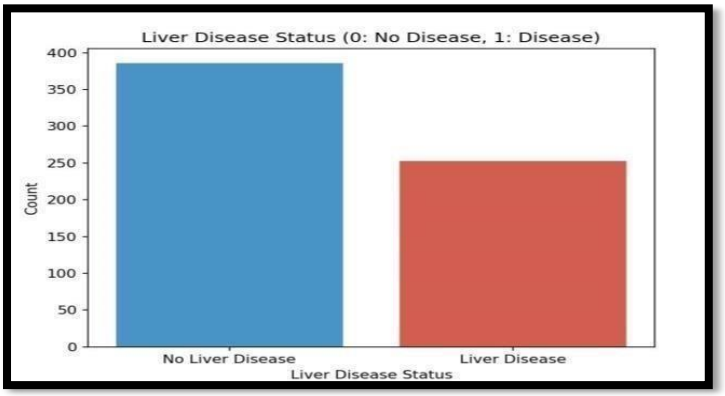
BLOCK DIAGRAM 3.2.2.2: THE STEPS FOR DATA PREPROCESSING AND DATA CLEANING

2. Missing Value Treatment: An examination of the dataset revealed missing values, especially within certain features. To maintain data completeness, missing values were imputed using appropriate strategies such as the median imputation technique. This approach mitigates the impact of missing information while retaining valuable data.

3.2.2.3. Exploratory Data Analysis (EDA)

EDA stands for Exploratory Data Analysis. It is an approach to analyzing data sets to summarize their main characteristics, often with the help of visual representations. EDA is a crucial initial step in data analysis and research, as it helps researchers and data analysts understand the data, identify patterns, relationships, anomalies, and trends within the dataset.
Liver Disease Distribution Analysis

The bar chart visually represents the distribution of patients in the dataset based on their liver disease status. On the x-axis, there are two categories: "No Liver Disease" (0) and "Liver Disease" (1). The y-axis indicates the count of patients belonging to each category. The blue bar corresponds to the number of patients with no liver disease (0), while the red bar represents the number of patients with liver disease (1). This bar chart provides a clear overview of how many patients have liver disease and how many do not, allowing for easy comparison of these two groups within the dataset. We created a count plot, which visually represents the distribution of patients based on their liver disease status. This plot helps us see the balance between patients with and without liver disease. It is an essential step in preparing our data for further analysis and modelling.



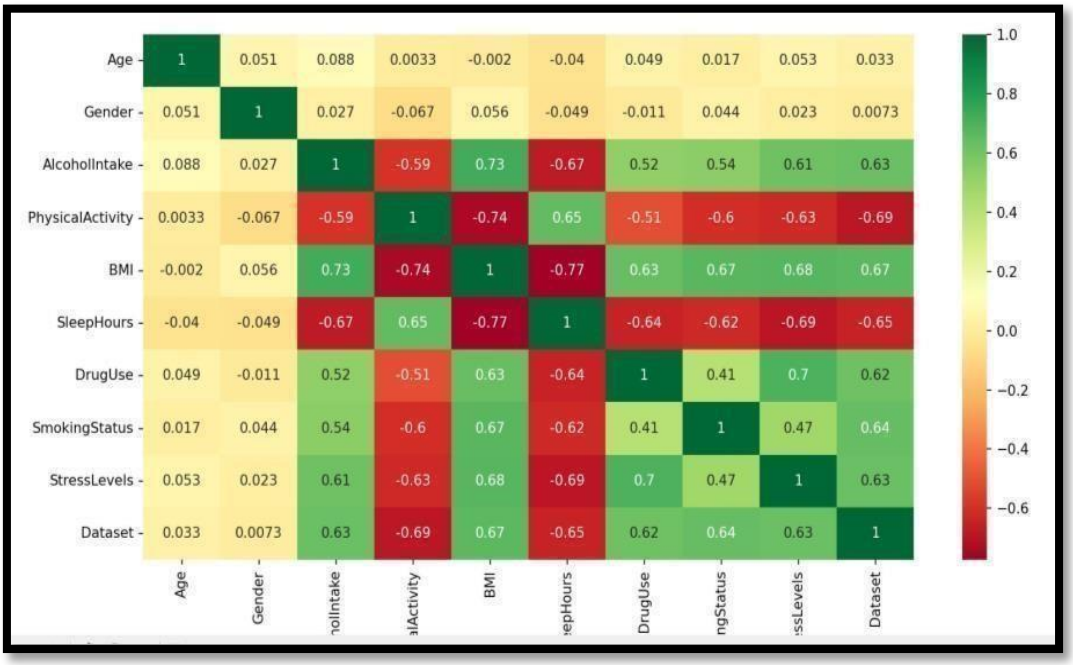
PLOT 3.2.2.3: THE DISTRIBUTION OF PATIENT DATA WITH AND WITH OUT LIVER DISEASE

3.2.2.4. Label Encoding

Label encoding is a method employed to convert categorical data, such as gender, alcohol consumption, smoking status, and other categorical attributes, into numerical values. This transformation facilitates the utilization of machine learning algorithms, as these algorithms often require numerical inputs. The label encoders are applied to several categorical columns, enabling the conversion of their categorical values into numerical representations. This preprocessing step enhances the compatibility of the data with machine learning models.

3.2.2.5. Feature Selection

Feature selection is the process of choosing a subset of the most relevant and informative features (variables or attributes) from a larger set of features in a dataset. The goal of feature selection is to improve model performance, reduce overfitting, and enhance the interpretability of machine learning models by retaining only the most valuable input variables.



PLOT 3.2.2.5: FEATURE SELECTION CORRELATION MATRIX

The heatmap output provides a visual representation of the correlations between different attributes in the dataset. It uses a color scale, where colors like green indicate positive correlations, red indicates negative correlations, and lighter shades suggest weaker or no correlations. The diagonal line in the heatmap is uniformly colored as it represents a feature's correlation with itself, which is always perfect. This heatmap helps you identify which features are strongly correlated and informs your feature selection process, allowing you to choose relevant features for further analysis and model building while avoiding multicollinearity issues. The correlations between different attributes in the dataset and selects the most relevant ones based on a threshold value. The selected features, such as 'Age,' 'Gender,' 'Alcohol Intake,' 'BMI,' 'Drug Use,' 'Smoking Status,' and 'Stress Levels,' are considered important for predicting liver disease ('Dataset').

3.2.2.6. Result

In our comparative analysis of three machine learning algorithms—Logistic Regression, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN)—for predicting liver disease based on lifestyle attributes and demographic data, we also examined their performance through confusion matrices.

Logistic Regression displayed an accuracy of 88.75%, with a precision of 85% and recall of 89%. In its confusion matrix, we observed 86 true negatives (correctly predicted "No Liver Disease") and 56 true positives (correctly predicted "Liver Disease"), along with 9 false negatives and 9 false positives.

Accuracy of the Logistic Regression Model is: 0.8875

```
[[86  9]
 [ 9 56]]
classification report:
              precision    recall  f1-score   support

           0       0.91      0.91      0.91        95
           1       0.86      0.86      0.86        65

    accuracy                           0.89       160
   macro avg       0.88      0.88      0.88       160
weighted avg       0.89      0.89      0.89       160
```

FIG3.2.2.6.1: CLASSIFICATION REPORT FOR LOGISTIC REGRESSION

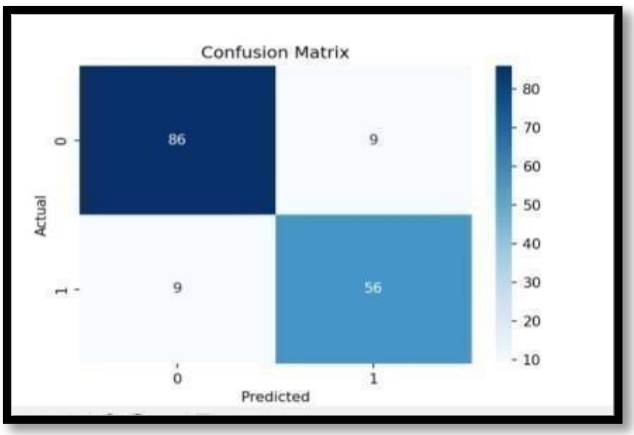


FIG3.2.2.6.2: CONFUSION MATRIX FOR LOGISTIC REGRESSION

Support Vector Machine (SVM) slightly outperformed the other models with an accuracy of 89.37%, precision of 85%, and recall of 89%. Its confusion matrix revealed 85 true negatives and 58 true positives, with 10 false negatives and 7 false positives.

```
Accuracy of the svm is : 0.89375
[[85 10]
 [ 7 58]]
classification report:
              precision    recall  f1-score   support

           0       0.92      0.89      0.91        95
           1       0.85      0.89      0.87        65

    accuracy                           0.89       160
   macro avg       0.89      0.89      0.89       160
weighted avg       0.90      0.89      0.89       160
```

FIG3.2.2.6.3: CLASSIFICATION REPORT FOR SVM

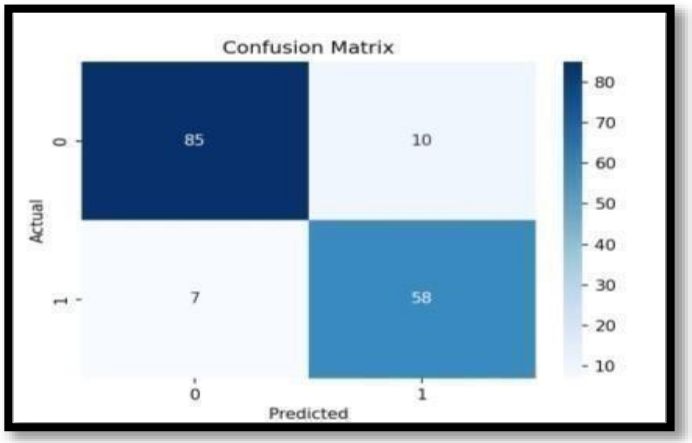


FIG3.2.2.6.4: CONFUSION MATRIX FOR SVM

K-Nearest Neighbors (KNN) achieved an accuracy of 88.12%, precision of 86%, and recall of 85%. In its confusion matrix, we observed 86 true negatives and 55 true positives, along with 9 false negatives and 10 false positives.

```
accuracy of knn 0.88125
[[86  9]
 [10 55]]
classification report:
              precision    recall  f1-score   support

           0       0.90      0.91      0.90        95
           1       0.86      0.85      0.85        65

    accuracy                           0.88       160
   macro avg       0.88      0.88      0.88       160
weighted avg       0.88      0.88      0.88       160
```

FIG3.2.2.6.5: CLASSIFICATION REPORT FOR KNN

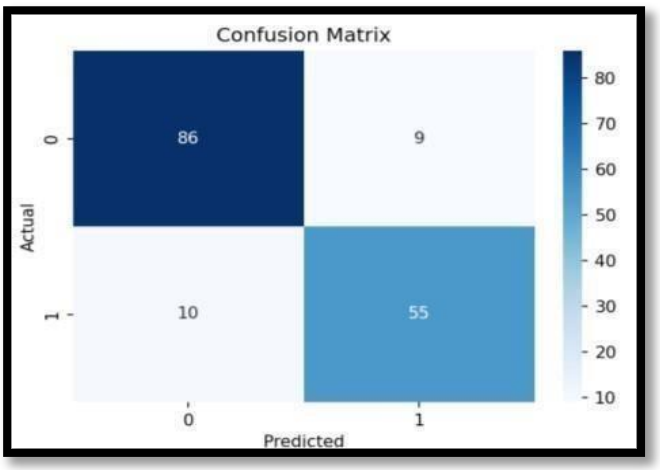


FIG3.2.2.6.4: CONFUSION MATRIX FOR KNN

After a comprehensive evaluation of three machine learning algorithms—Logistic Regression, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN)—it's evident that SVM emerges as the most robust and reliable model for liver disease prediction. With an impressive accuracy of 89.37%, SVM outperforms its counterparts. It excels in both precision and recall, making it the preferred choice for identifying individuals with or without liver disease. The SVM model demonstrates the highest capability in delivering accurate predictions and is thus the most suitable algorithm for this crucial medical application.

3.2.3. MODULES

3.2.3.1. Preprocessing Description

Real world data usually have the following drawbacks: Incompleteness, Noisy and Inconsistence. So, these data need to be preprocessed to get the data suitable for analysis purposes, and the preprocessing includes the following tasks:

- Data cleaning: fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies. Data integration: using multiple databases, data cubes, or files.
- Data transformation: normalization and aggregation.
- Data reduction: reducing the volume but producing the same or similar analytical results.
- Data discretization: part of data reduction, replacing numerical attributes with nominal ones.

3.2.3.1 Prediction Description

Prediction description in your project involves using machine learning models to estimate an individual's risk of developing liver disease based on their lifestyle attributes, such as diet, physical activity, stress levels, alcohol consumption, and smoking habits. These models analyze historical data to make predictions about whether an individual is at risk of liver disease, and the accuracy of these predictions is evaluated to assess the model's effectiveness in early disease detection.

- **Data Preparation:** You start by reading the dataset containing lifestyle attributes and liver disease status. The dataset is preprocessed to ensure data quality and consistency. This includes removing duplicate records and handling missing values. Categorical variables, such as gender, alcohol consumption, and smoking status, are encoded into numerical values to make them suitable for machine learning.
- **Feature Selection:** Feature selection is a crucial step in improving the predictive accuracy of machine learning models. You calculate the correlation between different attributes in the dataset to identify the most relevant features. The selected features are those that have a significant impact on predicting liver disease.
- **Model Selection:** You choose three different machine learning algorithms for liver disease prediction: Logistic Regression, Support Vector Machine (SVM), and K-nearest Neighbors (KNN). Each of these algorithms has its strengths and is suitable for early disease detection.
- **Training the Models:** For each of the selected machine learning algorithms, you split the data into training and testing sets. The training set is used to train the model on the historical data, while the testing set is used to evaluate the model's performance. The data is also standardized using the Standard Scaler module to ensure that features have a mean of 0 and a standard deviation of 1, which can help improve model performance.
- **Making Predictions:** Once the models are trained, they are applied to the testing data to make predictions about liver disease status. For each algorithm, you calculate the accuracy of the predictions using metrics like accuracy_score, and you create confusion matrices to visualize the performance of the models.
- **Evaluation:** The accuracy and performance of each model are evaluated to determine which algorithm is the most effective for liver disease prediction based on lifestyle attributes. The results are presented through visualizations like confusion matrices.

The prediction phase is a critical component of your project, as it determines the ability of the machine learning models to accurately assess liver disease risk based on lifestyle factors. The choice of multiple algorithms allows for a comprehensive evaluation of their performance and suitability for early disease detection.

Here are some specific modules that can be used with this algorithm.

Several Python modules and libraries can be used to enhance the functionality and performance of your machine learning algorithms. Here are some essential modules commonly used in machine learning projects:

- **NumPy**: NumPy is a fundamental library for numerical and matrix operations. It is used for data manipulation and array computations, making it a core component in many machine learning workflows.
- **Pandas:** Pandas is a versatile library for data manipulation and analysis. It provides data structures such as data frames that are crucial for handling datasets and performing data preprocessing.
- **Scikit-Learn (Sk learn):** Scikit-Learn is a comprehensive machine learning library that offers a wide range of algorithms for classification, regression, clustering, and more. You can use it for model training, evaluation, and hyperparameter tuning.
- **Matplotlib and Seaborn:** These libraries are used for data visualization. They help create various types of charts, graphs, and visualizations to better understand and communicate your results.
- **Scipy**: Scipy is an extension of NumPy and provides additional functionality for scientific and technical computing, including optimization, integration, and statistical functions.
- **Standard Scaler from sklearn preprocessing:** This module is used for feature scaling, ensuring that all features have a mean of 0 and a standard deviation of 1. It's important for improving the performance of certain machine learning algorithms.
- **Label Encoder from sklearn preprocessing:** Label Encoder is used to convert categorical data, such as gender, smoking status, or alcohol consumption, into numerical values, making it compatible with machine learning algorithms.
- **K-Neighbors Classifier, Logistic Regression, and SVC from sklearn:** These are the specific machine learning algorithms you have used in your project for binary classification. You can explore other classifiers available in Scikit-Learn as well.
- **Job lib or Pickle:** These libraries are used for model persistence, allowing you to save and load trained machine learning models for future use.
- **Statistics:** Python's built-in statistics library can be useful for calculating various statistical metrics to evaluate the performance of your models, such as precision, recall, F1-score, and more.

3.3. SYSTEM ANALYSIS METHODS

3.3.1. USE CASE DIAGRAM

Use case diagram represent the overall scenario of the system. A scenario is nothing but a sequence of steps describing an interaction between a user and a system. Thus, use case is a set of scenario tied together by some goal. The use case diagram is drawn for exposing the functionalities of the system.

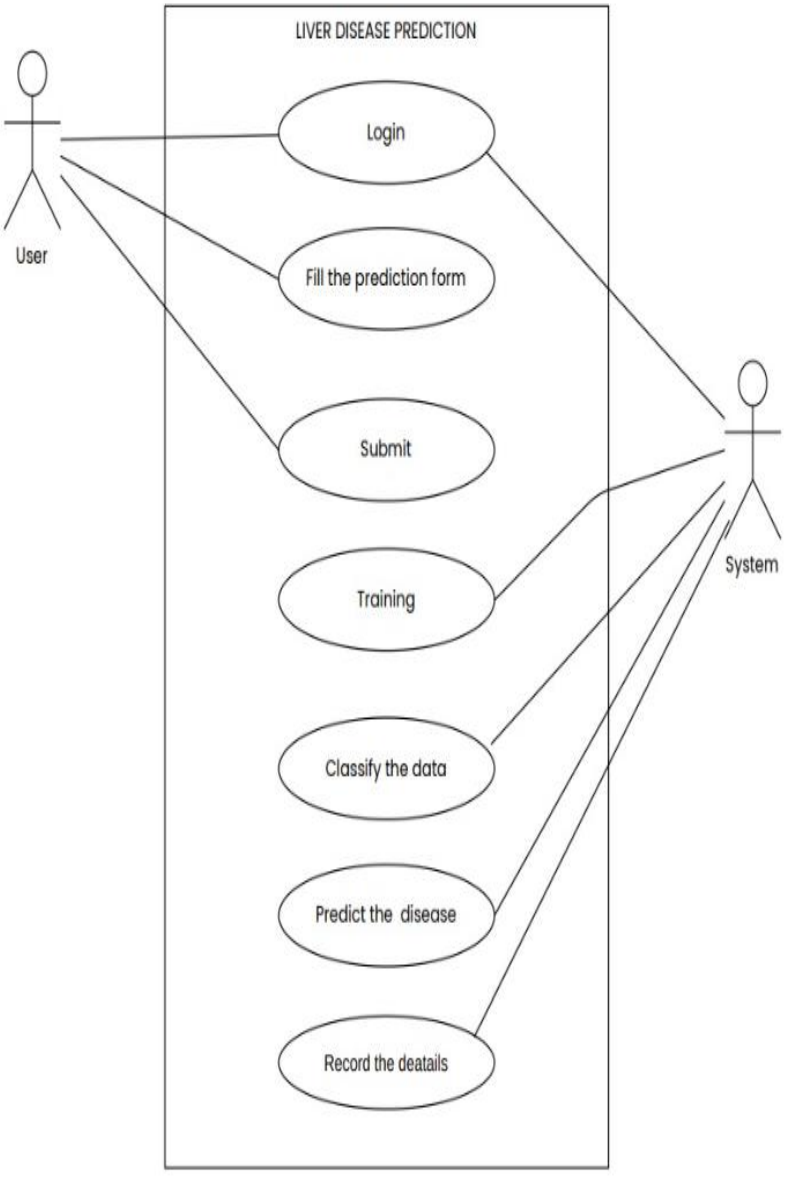


Fig: 3.3.1. USE CASE DIAGRAM

3.3.2. DATAFLOW DIAGRAM

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation. Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business.

The physical data flow diagram describes the implementation of the logical data flow.
Steps Involved In Data Flow Diagram of Our System

1. Start the process loading the dataset.
2. Commence the process by collecting a dataset of facial images labeled with corresponding emotions.
3. Preprocess the collected data, including resizing images to a standard size, normalizing pixel values, and augmenting the dataset for diversity.
4. Select a deep learning model, such as Mini-Xception, specialized for facial emotion recognition.
5. Train the chosen model on the preprocessed dataset to recognize patterns in facial expressions linked to various emotions.
6. Evaluate the trained model's performance using a testing dataset, calculating metrics like accuracy and loss.
7. Utilize the trained model for real-time video analysis, capturing live video streams from a webcam.
8. Classify emotions based on the model's predictions, including categories like anger, happiness, sadness, and surprise.
9. Offer a user-friendly interface for real-time display of analyzed emotions, potentially with graphical representations or textual labels.
10. Terminate the processs.

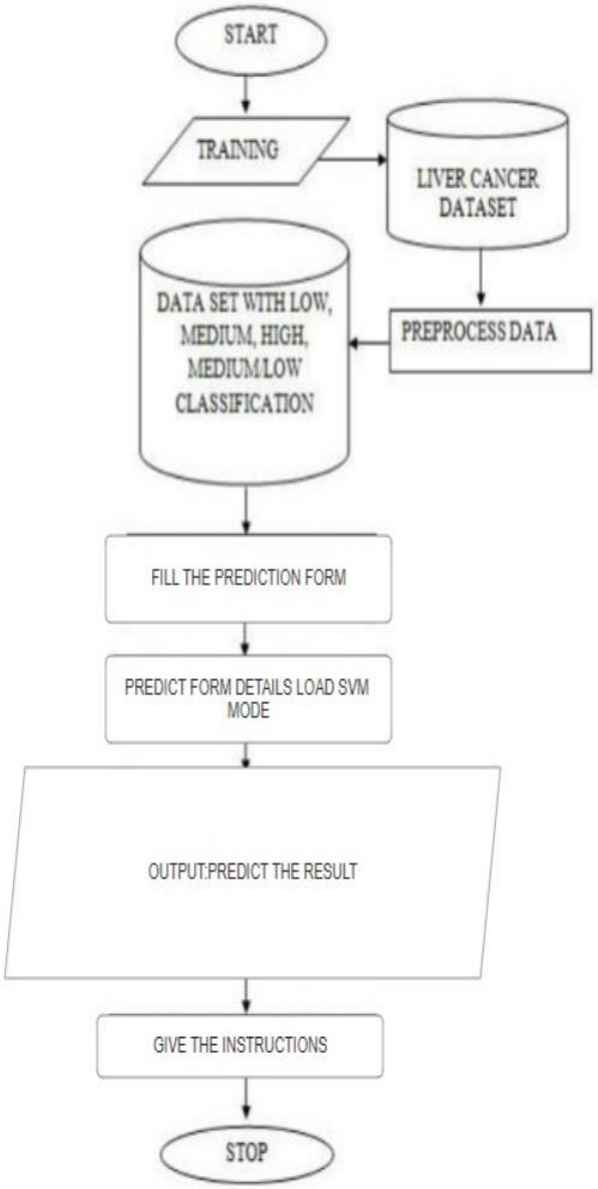


Fig: 3.3.2. DATA FLOW DIAGRAM

3.3. SYSTEM DESIGN

3.3.1. SYSTEM ARCHITETURE

The purpose of the design phase is to plan a solution of the problem specified by the requirement document. This phase is the first step in moving from the problem domain to the solution domain. In other words, starting with what is needed, design takes us toward how to satisfy the needs. The design of a system is perhaps the most critical factor affection the quality of the software; it has a major impact on the later phase, particularly testing, maintenance. The output of this phase is the design document. This document is similar to a blueprint for the solution and is used later during implementation, testing and maintenance. The design activity is often divided into two separate phases System Design and Detailed Design.

System Design also called top-level design aims to identify the modules that should be in the system, the specifications of these modules, and how they interact with each other to produce the desired results. At the end of the system design all the major data structures, file formats, output formats, and the major modules in the system and their specifications are decided.

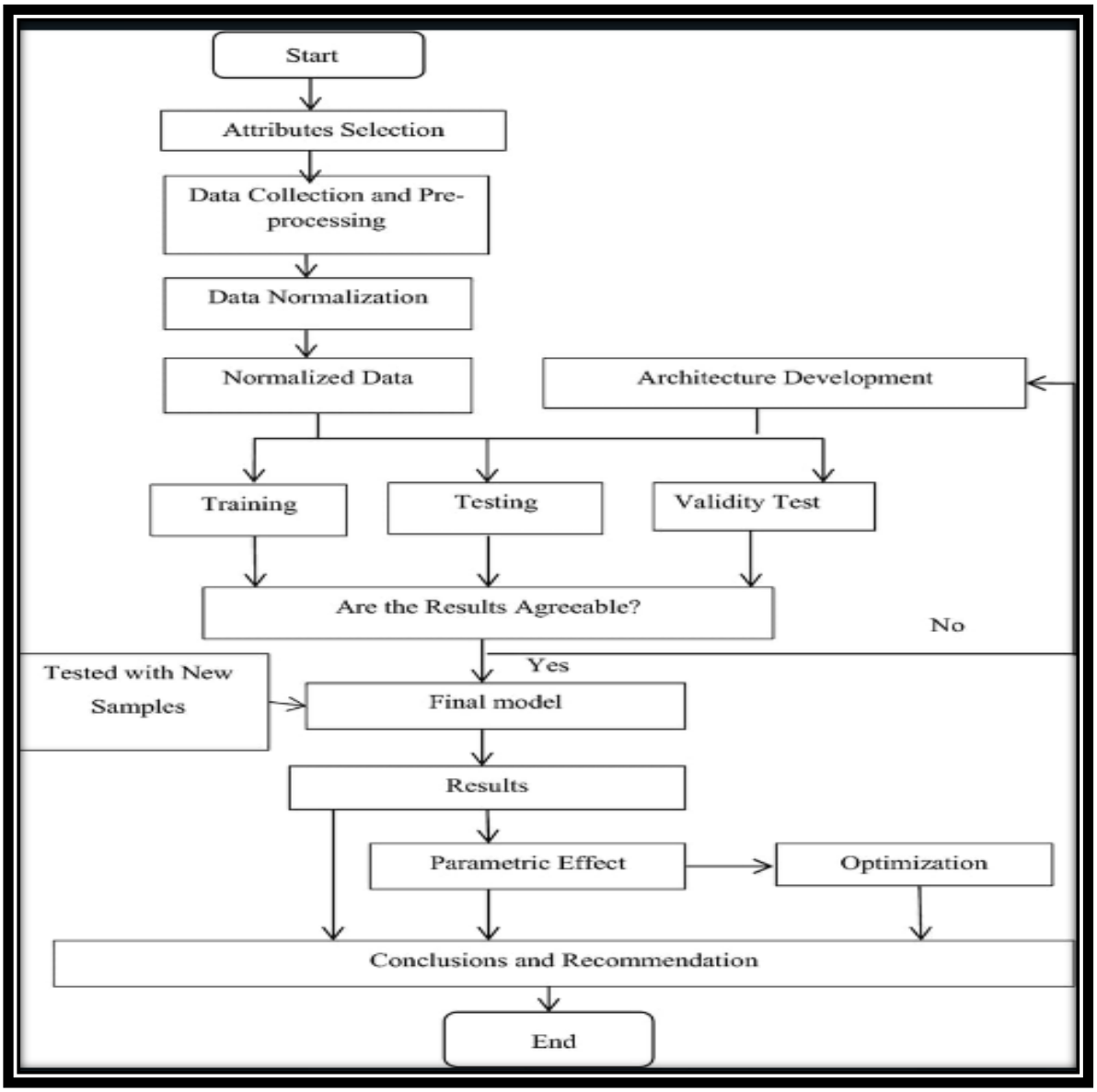


Fig 3.3.1 System Architecture

3.3.2. CLASS DIAGRAM

Class Diagram gives the static view of an application. A class diagram describes the types of objects in the system and the different types of relationships that exist among them. This modeling method can run with almost all Object- Oriented Methods. A class can refer to another class. A class can have its objects or may inherit from other classes.

- Class Diagram Illustrates data models for even very complex information systems
- It provides an overview of how the application is structured before studying the actual code. This can easily reduce the maintenance time
- It helps for better understanding of general schematics of an application.
- Allows drawing detailed charts which highlights code required to be programmed
- Helpful for developers and other stakeholders

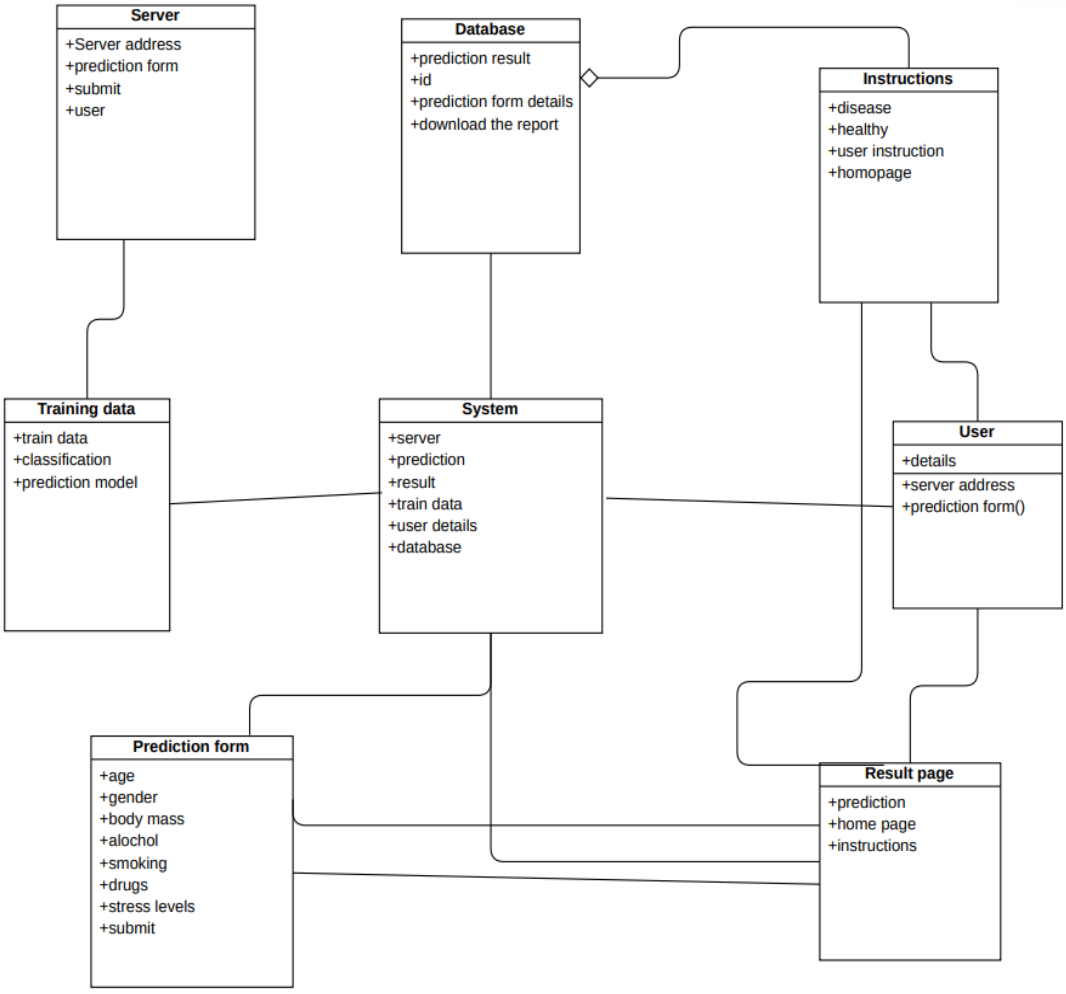


Fig:3.3.2. CLASS DIAGRAM

CHAPTER – 4

IMPLEMENTATION

4. IMPLEMENTATION

4.1. TOOLS USED

INTRODUCTION TO PYTHON

Python is a high-level, interpreted, interactive and object- oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- Python is Interpreted: Python is processed at run time by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive: You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object-Oriented: Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- Python is a Beginner's Language: Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

- Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands
- Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol68, Small Talk, Unix shell, and other scripting languages.
- Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).
- Python is now maintained by a core development team the institute, although Guido van Rossum still holds a vital role in directing its progress.
- Python’s standard library: Pandas, Numpy, Sklearn, Matplotlib, Tensorflow, cv2, PLI Importing Datasets

PANDAS

Pandas is quite a game changer when it comes to analyzing data with Python and it is one of the most preferred and widely used tools in data munging/wrangling if not THE most used one. Pandas is an open source. What’s cool about Pandas is that it takes data (like a CSV or TSV file, or a SQL database) and creates a Python object with rows and columns called data frame that looks very similar to table in a statistical software (think Excel or SPSS for example. People who are familiar with R would see similarities to R too). This is so much easier to work with in comparison to working with lists and/or dictionaries through for loops or list comprehension.

NUMPY

Numpy is one such powerful library for array processing along with a large collection of high- level mathematical functions to operate on these arrays. These functions fall into categories like Linear Algebra, Trigonometry, Statistics, Matrix manipulation, etc. Getting NumPy NumPy's main object is a homogeneous multidimensional array. Unlike python's array class which only handles one dimensional array, NumPy's and array class can handle multidimensional array and provides more functionality. NumPy's dimensions are known as axes. For example, the array below has 2 dimensions or 2 axes namely rows and columns. Sometimes dimension is also known Page 58 as a rank of that particular array or matrix.

MATPLOTLIB

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hard copy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

SCIKIT-LEARN

Scikit-Learn, often referred to as sklearn, is a versatile and widely-used open-source machine learning library in Python. It serves as a fundamental tool for data scientists, researchers, and developers in building and implementing machine learning models and data analysis workflows. Scikit-Learn provides a rich set of tools, algorithms, and functions that streamline the machine learning process.

At its core, scikit-learn is designed to be user-friendly and efficient, making it an excellent choice for both beginners and experienced machine learning practitioners. It offers a unified interface for various machine learning tasks, including classification, regression, clustering, dimensionality reduction, model selection, and data preprocessing.

One of the stand out features of scikit-learn is its well-documented, consistent API, which simplifies the process of training and evaluating models. This uniformity allows users to easily swap and test different algorithms and techniques while maintaining a consistent workflow.

Scikit-learn also provides an array of supervised and unsupervised learning algorithms, such as linear regression, decision trees, support vector machines, k-means clustering, and many more. The library includes tools for feature selection, data transformation, and model evaluation, making it a one-stop solution for machine learning tasks.

Furthermore, scikit-learn is highly compatible with other popular Python libraries, like NumPy and Pandas, facilitating seamless data manipulation and integration into existing data science pipelines. Its commitment to simplicity, robustness, and performance has made scikit-learn a trusted resource for a broad range of applications, from data exploration and model development to real-world deployments in fields like healthcare, finance, and natural language processing.

In summary, scikit-learn empowers data scientists and machine learning practitioners by providing a versatile and accessible framework for building, evaluating, and deploying machine learning models, making it a fundamental part of the Python data science ecosystem.

SCIPY

SciPy, a critical scientific computing library in Python, extends the capabilities of NumPy by offering an extensive range of tools for scientific and technical computing. It includes specialized modules for optimization, integration, linear algebra, signal processing, and more. Its unified API simplifies complex scientific computations, and it seamlessly integrates with libraries like NumPy and Matplotlib. SciPy is an essential resource for researchers and scientists in various fields, enabling them to perform advanced mathematical and scientific tasks with efficiency and simplicity.

4.2. PSEUDO CODE

Python File:

```
#model.py accuracy
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import accuracy_score
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, mean_absolute_error,r2_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
import pickle
#READ CSV FILE
data= pd.read_csv("D:\\programs\\5a1\\new\\liver12.csv")
print(data.head())
print(data.shape)
print(data.columns)
#DATA CLEANING
print(data.duplicated())
print(data.duplicated().sum())
print(data.drop_duplicates())
print(data.shape)
#CHECKING MISSING VALUES
print(data.isna().sum())
missing_values = data.isnull().sum()
numeric_columns = data.select_dtypes(include=['number']).columns
print("The numeric coumnns are:",numeric_columns)
for column in numeric_columns:
    data[column].fillna(data[column].mean(), inplace=True)
missing_values_after = data.isnull().sum()
print(missing_values_after)
# Remove leading and trailing spaces from column names
data.columns = data.columns.str.strip()
#FEATURE SELECTION
corrmat = data.corr()
top_corr_features = corrmat.index
plt.figure(figsize=(10,10))
g=sns.heatmap(data[top_corr_features].corr(),annot=True,cmap="RdYlGn")
#plt.show()
threshold = 0.5
selected_features = corrmat.columns[corrmat.abs().mean() >threshold]
print("features are:", selected_features )
X = data[['Age', 'Gender', 'AlcoholIntake','BMI','DrugUse','SmokingStatus','StressLevels']]
y = data['Dataset']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,random_state=42)
sc=StandardScaler()
X_train_std=sc.fit_transform(X_train)
X_test_std = sc.transform(X_test)
svm_model = SVC(kernel='linear')
```

```
# Train the SVM model on the training data
svm_model.fit(X_train_std, y_train)

y_pred = svm_model.predict(X_test_std)
# Make predictions on the test data
from sklearn.metrics import classification_report
classification_rep = classification_report(y_test, y_pred)

print("Classification Report svm:\n", classification_rep)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy of the svm is :", accuracy)

cm = confusion_matrix(y_test, y_pred)
print(cm)
#LogisticRegression
from sklearn.linear_model import LogisticRegression
logreg=LogisticRegression()
logreg.fit(X_train_std, y_train)
y_predlog=logreg.predict(X_test_std)
from sklearn.metrics import accuracy_score
classification_rep = classification_report(y_test, y_predlog)
print("Classification Report LogisticRegression: \n", classification_rep)
accuracy1og = accuracy_score(y_test, y_predlog)
print("Accuracy of the LogisticRegression is :", accuracy1og)
#knn
from sklearn.neighbors import KNeighborsClassifier
knn_classifier = KNeighborsClassifier(n_neighbors=21, metric='euclidean')
knn_classifier.fit(X_train_std,y_train)
knn_y_pred=knn_classifier.predict(X_test_std)
cm=confusion_matrix(y_test,knn_y_pred)
print(cm)
classification_rep = classification_report(y_test, knn_y_pred)
print("Classification Report knn:\n", classification_rep)
accuracyknn = accuracy_score(y_test, knn_y_pred)
print("Accuracy of the KNeighborsClassifier is :", accuracyknn)
# Classification Report and Bar Graph for SVM
classification_rep_svm = classification_report(y_test, y_pred, output_dict=True)
class_names = [str(cls) for cls in classification_rep_svm if cls != 'accuracy' and cls != 'macro avg']
precision_svm = [classification_rep_svm[cls]['precision'] for cls in class_names]
recall_svm = [classification_rep_svm[cls]['recall'] for cls in class_names]
f1_score_svm = [classification_rep_svm[cls]['f1-score'] for cls in class_names]

plt.figure(figsize=(10, 6))
x = range(len(class_names))
plt.bar(x, precision_svm, width=0.3, label='Precision', align='center')
plt.bar(x, recall_svm, width=0.3, label='Recall', align='edge')
plt.bar(x, f1_score_svm, width=0.3, label='F1-Score', align='edge')
plt.xlabel('Class')
plt.ylabel('Score')
plt.title('Classification Report Metrics (SVM)')
plt.xticks(x, class_names)
plt.legend(loc='best')
plt.tight_layout()
```

```
plt.show()
# Classification Report and Bar Graph for Logistic Regression
classification_rep_logreg = classification_report(y_test, accuracy1og, output_dict=True)
class_names = [str(cls) for cls in classification_rep_logreg if cls != 'accuracy' and cls != 'macro avg']
precision_logreg = [classification_rep_logreg[cls]['precision'] for cls in class_names]
recall_logreg = [classification_rep_logreg[cls]['recall'] for cls in class_names]


f1_score_logreg = [classification_rep_logreg[cls]['f1-score'] for cls in class_names]

plt.figure(figsize=(10, 6))
x = range(len(class_names))
plt.bar(x, precision_logreg, width=0.2, label='Precision', align='center')
plt.bar(x, recall_logreg, width=0.2, label='Recall', align='edge')
plt.bar(x, f1_score_logreg, width=0.2, label='F1-Score', align='edge')
plt.xlabel('Class')
plt.ylabel('Score')
plt.title('Classification Report Metrics (Logistic Regression)')
plt.xticks(x, class_names)
plt.legend(loc='best')
plt.tight_layout()

plt.show()

# Classification Report and Bar Graph for K-Nearest Neighbors
classification_rep_knn = classification_report(y_test, knn_y_pred, output_dict=True)
class_names = [str(cls) for cls in classification_rep_knn if cls != 'accuracy' and cls != 'macro avg']
precision_knn = [classification_rep_knn[cls]['precision'] for cls in class_names]
recall_knn = [classification_rep_knn[cls]['recall'] for cls in class_names]
f1_score_knn = [classification_rep_knn[cls]['f1-score'] for cls in class_names]

plt.figure(figsize=(10, 6))
x = range(len(class_names))
plt.bar(x, precision_knn, width=0.2, label='Precision', align='center')
plt.bar(x, recall_knn, width=0.2, label='Recall', align='edge')
plt.bar(x, f1_score_knn, width=0.2, label='F1-Score', align='edge')
plt.xlabel('Class')
plt.ylabel('Score')
plt.title('Classification Report Metrics (K-Nearest Neighbors)')
plt.xticks(x, class_names)
plt.legend(loc='best')
plt.tight_layout()

plt.show()

# Save the trained SVM model to a pickle file
with open("model.pkl", "wb") as model_file:
    pickle.dump(svm_model, model_file)

#app.py flask
import pickle
import numpy as np
from flask import Flask, render_template, request, send_file  # Import 'send_file' for serving files
import pymongo

app = Flask(__name__)
```

```
# Load the trained machine learning model
with open('model.pkl', 'rb') as model_file:
    model = pickle.load(model_file)

# Connect to MongoDB
mongo_client = pymongo.MongoClient("mongodb://localhost:27017/")  # Replace with your MongoDB
connection URI
db = mongo_client["liver"]
collection = db["liverpre"]


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

@app.route('/index', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        return render_template('index.html', prediction_text="")
    else:
        return render_template('index.html')

@app.route('/instruction', methods=['GET', 'POST'])
def instruction():
    if request.method == 'POST':
        return render_template('instruction.html', prediction_text="")
    else:
        return render_template('instruction.html')

@app.route('/about', methods=['GET', 'POST'])
def about():
    if request.method == 'POST':
        return render_template('about.html', prediction_text="")
    else:
        return render_template('about.html')

@app.route('/feedback', methods=['GET', 'POST'])
def feedback():
    if request.method == 'POST':
        return render_template('response.html')
    else:
        return render_template('feedback.html')

@app.route('/predict', methods=['GET', 'POST'])
def predict():
    if request.method == 'POST':
        # Get user input from the form
        Age = float(request.form['Age'])
        Gender = int(request.form['Gender'])
        AlcoholIntake = float(request.form['AlcoholIntake'])
        BMI = float(request.form['BMI'])
        DrugUse = int(request.form['DrugUse'])
        SmokingStatus = float(request.form['SmokingStatus'])
        StressLevels = float(request.form['StressLevels'])
```

```
        # Preprocess the user input (scaling, etc.) - same preprocessing as your training data
        input_data = np.array([Age, Gender, AlcoholIntake, BMI, DrugUse, SmokingStatus,
StressLevels]).reshape(1, -1)

        # Make a prediction using the loaded SVM model
        prediction = model.predict(input_data)

        # Display the prediction (you can modify this part to display it as needed)
        if prediction[0] == 1:
            prediction_text = "Liver Disease Detected"
        else:
            prediction_text = "No Liver Disease Detected"

    # Save the user input and prediction to MongoDB
        user_data = {
            "Age": Age,
            "Gender": Gender,
            "AlcoholIntake": AlcoholIntake,
            "BMI": BMI,
            "DrugUse": DrugUse,
            "SmokingStatus": SmokingStatus,
            "StressLevels": StressLevels,
            "Prediction": prediction_text
        }

        # Insert the user data into the MongoDB collection
        collection.insert_one(user_data)

        try:
            return render_template('result.html', prediction_text=prediction_text)
        except Exception as e:
            print(f"Error: {str(e)}")
            return "An error occurred while making predictions."

@app.route('/download_pdf')  # Define a new route for downloading the PDF
def download_pdf():
    # Replace 'path_to_your_pdf.pdf' with the actual path to your PDF file
    pdf_path = 'C:\\Users\\manik\\Desktop\\new'
    return send_file(pdf_path, as_attachment=True, download_name='filename.pdf')

if __name__ == '__main__':
    app.run(debug=True)

#home.html
<!DOCTYPE html>
<html>

<head>
    <title>LIVER DISEASE PREDICTION</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #e5e7e3;
            padding: 0;
```

```
 text-align: center;
        }
        p {
            color: #666;
            font-size: 20px;
            margin-bottom: 20px;
        }

    </style>
</head>

<body>
    <center>
        <h1>LIVER DISEASE PREDICTION</h1>
    </center>

 <hr size="3" color="black">
    <table width="90%">

 <tr>
            <td><a href="{{url_for('home')}}">HOME</a></td>
            <td><a href="{{url_for('index')}}">PREDICTION FORM</a></td>
            <td><a href="{{url_for('instruction')}}">INSTRUCTIONS</a></td>
            <td><a href="{{url_for('about')}}">ABOUT</a></td>
            <td><a href="{{url_for('feedback')}}">FEEDBACK</a></td>
        </tr>
    </table>

    <img
src="https://th.bing.com/th/id/R.db0de5e05ffc18771a564aa493829b0f?rik=05UPqDHKkdIy0Q&riu=http%3a%2f%2fpeoplebeatingcancer.org%2fwp-content%2fuploads%2f2018%2f04%2fliver-image.jpg&ehk=LCvPnT7yVBVgYyhHE1e2dWRXufNYFr86oO5UsM9E%2fSs%3d&risl=&pid=ImgRaw&r=0" alt="Liver Image"
        align="center" width="400" height="300">
    <hr size="3" color="white">
    <marquee style="color:rgb(238, 234, 234)" width="100%" direction="right" bgcolor="green"
height="20px">LIVER DISEASE PREDICTION</marquee>
    <h1>Welcome to Liver Disease Prediction</h1>
    <p>This website is dedicated to helping you predict the likelihood of liver disease based on your health data.</p>
    <p>Explore our tools and resources to better understand your health and well-being.</p>

</body>

</html>
#prediction.html

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Liver Disease Prediction</title>
```

```
<style>
        /* Reset some default styles */
        body, h1, h2, p, form {
            margin: 0;
            padding: 0;
        }

        /* Basic styling */
        body {
            font-family: Arial, sans-serif;
            background-color: #f0f0f0;
        }

        header {
            background-color: #333;
            color: white;
            text-align: center;
            padding: 10px;
        }

        main {
            max-width: 800px;

margin: 0 auto;
            padding: 20px;
        }

        .prediction-form {
            background-color: white;
            border-radius: 5px;
            padding: 20px;
            box-shadow: 0px 0px 5px rgba(0, 0, 0, 0.2);
        }

        form label {
            display: block;
            margin-bottom: 10px;
        }

        form input {
            width: 100%;
            padding: 8px;
            margin-bottom: 10px;
            border: 1px solid #ccc;
            border-radius: 3px;
        }
       form select{
            width: 102%;
            padding: 8px;
            margin-bottom: 10px;
            border: 1px solid #ccc;
```

```
 border-radius:3px;}
      form button {
         background-color: #333;


         color: white;
         padding: 10px 15px;
         border: none;
         border-radius: 3px;
         cursor: pointer;
      }

      footer {
         text-align: center;
         padding: 10px;
         background-color: #333;
         color: white;
      }
   </style>
</head>
<body>
   <header>
      <h1>Liver Disease Prediction</h1>
   </header>
   <main>
      <div class="prediction-form">
         <form action="/predict" method="POST">
            <label for="Age">Age:</label>
            <input type="number" id="Age" name="Age" required><br>

            <label for="Gender">Gender:</label>
            <select id="Gender" name="Gender">
               <option value="0">Female</option>


               <option value="1">Male</option>
            </select><br>

            <label for="AlcoholIntake">Alcohol Intake:</label>
            <select id="AlcoholIntake" name="AlcoholIntake">
               <option value="0">No Intake</option>
               <option value="1">Low Intake</option>
               <option value="2">Moderate Intake</option>
               <option value="3">High Intake</option>
               <option value="4">Very High Intake</option>
            </select><br>
            <label for="BMI">Body Mass:</label>
            <input type="text" id="BMI" name="BMI" required><br>

            <label for="DrugUse">Drug Usage:</label>
            <select id="DrugUse" name="DrugUse">
               <option value="0">NO</option>
               <option value="1">YES</option>
            </select><br>
```

```
            <label for="SmokingStatus">Smoking Status:</label>
            <select id="SmokingStatus" name="SmokingStatus">
               <option value="0">NO</option>
               <option value="1">YES</option>
            </select><br>

            <label for="StressLevels">Stress Levels:</label>
            <select id="StressLevels" name="StressLevels">
               <option value="0">Low</option>
               <option value="1">High</option>
               <option value="0.5">Average</option>
            </select><br>

            <input type="submit" value="Predict">
         </form>
      </div>
   </main>
   <footer>
       liver disease prediction
   </footer>
</body>
</html>
#instruction.html
<!DOCTYPE html>
<html>

<head>
   <title>Instructions</title>
   <style>
      body {
         font-family: Arial, sans-serif;
         background-color: #e5e7e3;
         padding: 20px;
         text-align: center;
      }

      .container {

 max-width: 800px;
         margin: 0 auto;

         padding: 20px;
         background-color: #fff;
         border-radius: 5px;
         box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
      }

      h1 {
         color: #333;
      }
      a {
         display: block;
         margin-top: 20px;
```

```
 text-decoration: underline;
            color: #4219cb;
        }

        p {
            color: #666;
        }


 ul {
            list-style-type: disc;
            margin-left: 20px;
            color: #666;
        }
    </style>
</head>

<body>
    <div class="container">
        <h1>Instructions for Liver Disease Prediction</h1>
        <p>Welcome to the Liver Disease Prediction application. Here are some instructions on how to use the app:</p>

        <h2>Step 1: Fill in the Prediction Form</h2>
        <p>Click on the "Prediction Form" link in the navigation menu. You will be redirected to a form where you can enter your personal information and health data.</p>

        <h2>Step 2: Submit Your Data</h2>
        <p>After filling in the required information, click the "Predict" button. The application will analyze your data and provide a prediction regarding the presence of liver disease.</p>

        <h2>Step 3: View the Results</h2>
        <p>The prediction results will be displayed on the screen. The app will inform you whether liver disease is detected or not based on the provided data.</p>

        <h2>Step 4: About the App</h2>
        <p>If you want to learn more about this application and the team behind it, you can click on the "About" link in the navigation menu.</p>

        <h2>Step 5: Additional Information</h2>
        <p>For additional information, references, and resources related to liver disease, check the "Resources" section of the app.</p>

        <h2>Step 6: Download PDF Report</h2>
        <p>If you need a PDF report of your prediction results, you can use the "Download PDF" feature in the navigation menu.</p>


        <p>Thank you for using our Liver Disease Prediction application. If you have any questions or need further assistance, please don't hesitate to contact us.</p>
    </div>
    <p><a href="http://127.0.0.1:5000">Go back to the homepage</a></p>
</body>
</html>
```

#result.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Liver Disease Prediction Result</title>
    <style>
        body {

 font-family: Arial, sans-serif;
            background-color: #e0e1dd;
            margin: 0;
            padding: 0;
            text-align: center;
        }
        h1 {
            color: #333;
        }
        p {
            font-size: 28px;
        }
        a {
            display: block;
            margin-top: 20px;
            text-decoration: underline;
            color: #4219cb;
        }
        .header-image {
            display: block;
            margin: 0 auto;
            width: 300px; /* Adjust the width as needed */
        }
        /* Style for the prediction text */
        .prediction-text {
            font-size: 24px;
            font-weight: bold;
            color: hsl(208, 100%, 60%); /* Change the color to your preference */
        }
    </style>
    <marquee style="color:white" width="100%" direction="right" bgcolor="green" height="20px">
LIVER DISEASE PREDICTION </marquee>
</head>
<body>
    <img class="header-image"
src="https://th.bing.com/th/id/R.db0de5e05ffc18771a564aa493829b0f?rik=05UPqDHKkdIy0Q&riu=http
%3a%2f%2fpeoplebeatingcancer.org%2fwp-content%2fuploads%2f2018%2f04%2fliver-
image.jpg&ehk=LCvPnT7yVBVgYyhHE1e2dWRXufNYFr86oO5UsM9E%2fSs%3d&risl=&pid=ImgR
aw&r=0" alt="Header Image">
    <table width="100%">
        <tr>
```

```
 <td><a href="http://127.0.0.1:5000">Go back to the homepage</a></td>
      <td><a href="/download_pdf" download="filename.pdf">Download PDF</a></td>
      <td><a href="{{url_for('feedback')}}">Feedback</a></td>
   </tr>
   </table>
   {% if prediction_text == "No Liver Disease Detected" %}
   <p class="prediction-text" style="background-color: white;">{{ prediction_text }}</p>

   <p style="text-align: left;">Instructions:</p>
<p style="text-align: left; width: 25%; background-color: white; padding: 10px;">Based on the current assessment, no liver disease has been detected. However, it is essential to maintain a healthy lifestyle and continue regular medical check-ups to ensure ongoing liver health.</p>
   {% else %}
   <p class="prediction-text">{{ prediction_text }}</p>
   <p style="text-align: left;">Instructions:</p>
   <p style="text-align: left;">Please monitor this individual for any signs or symptoms of liver disease and conduct regular liver function tests.</p>
   {% endif %}

</body>
</html>
#feedback.html
<!DOCTYPE html>
<html>

<head>
   <title>Feedback</title>
   <style>
      body {
         font-family: Arial, sans-serif;
         background-color: #e5e7e3;
         padding: 20px;
         text-align: center;
      }

      .container {
         max-width: 800px;
         margin: 0 auto;
         padding: 20px;
         background-color: #fff;
         border-radius: 5px;
         box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
      }

      h1 {
         color: #333;
      }

      p {
         color: #666;
         font-size: 20px;
         margin-bottom: 20px;
      }
```

```
 form {
            margin: 20px 0;
        }

        label {
            font-weight: bold;
        }

 input[type="text"],
        textarea {
            width: 100%;
            padding: 10px;
            margin: 5px 0;
            border: 1px solid #ccc;
            border-radius: 3px;
        }

        input[type="submit"] {
            background-color: #007bff;
            color: #fff;
            padding: 10px 20px;
            border: none;
            border-radius: 3px;
            cursor: pointer;
        }
    </style>
</head>

<body>
    <div class="container">
        <h1>Feedback</h1>
        <p>We value your feedback! Please share your thoughts, suggestions, or comments with us.</p>

        <form action="/feedback" method="post">
            <label for="name">Your Name:</label>
            <input type="text" id="name" name="name" required>

            <label for="email">Your Email:</label>
            <input type="text" id="email" name="email" required>

            <label for="message">Your Feedback:</label>
            <textarea id="message" name="message" rows="4" required></textarea>

            <input type="submit" value="Submit Feedback">
        </form>
    </div>
</body>

</html>
#about.html
<!DOCTYPE html>
<html>
```

```
<head>
   <title>About Liver Disease Prediction</title>
   <style>
      body {
         font-family: Arial, sans-serif;
         background-color: #e5e7e3;
         padding: 20px;
         text-align: center;
      }
      .container {
         max-width: 800px;
         margin: 0 auto;
         padding: 20px;
         background-color: #fff;
         border-radius: 5px;
         box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
      }
      a {
         display: block;
         margin-top: 20px;
         text-decoration: underline;
         color: #4219cb;
      }

      h1 {
         color: #333;
      }

      p {
         color: #666;
      }
   </style>
</head>

<body>
   <div class="container">
      <h1>About Liver Disease Prediction</h1>
      <p>Welcome to the Liver Disease Prediction application. This application is designed to assist in the early detection of liver disease and provide users with valuable insights into their health.</p>

      <h2>Our Mission</h2>
      <p>Our mission is to make healthcare more accessible and empower individuals to take control of their well-being. We believe that early detection and awareness can greatly improve health outcomes.</p>

      <h2>How It Works</h2>
      <p>The Liver Disease Prediction app uses machine learning to analyze user-submitted health data and provides predictions regarding the presence of liver disease. It leverages a trained model based on real-world medical data and research.</p>

      <h2>Privacy and Data Security</h2>
      <p>We take user privacy and data security seriously. Your personal and health information is kept confidential and is not shared with any third parties. We adhere to strict privacy policies and security measures to protect your data.</p>
```

```
 <h2>Contact Us</h2>
        <p>If you have any questions, feedback, or concerns, please feel free to contact us. We are here to assist you and provide support for using the Liver Disease Prediction app.</p>

        <p>Thank you for using our application. We hope it helps you on your journey to better health and well-being.</p>
    </div>
    <p><a href="http://127.0.0.1:5000">Go back to the homepage</a></p>
</body>
</html>


#response.html
<!DOCTYPE html>
<html>

<head>
    <title>Thank You</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #e5e7e3;
            padding: 20px;
            text-align: center;
        }

        .container {
            max-width: 800px;
            margin: 0 auto;
            padding: 20px;
            background-color: #efeaea;
            border-radius: 5px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
        }

        h1 {
            color: #333;
        }

        p {
            color: #666;
            font-size: 20px;
            margin-bottom: 20px;
        }

        /* Additional styles for the thank you message */
        .thank-you {
            font-size: 24px;
            color: #0040ff;
            font-weight: bold;
        }
```

```
  /* Style for links */
      a {
         text-decoration: underline;
         color: #0040ff;
         font-weight: bold;
         margin-top: 70px;
         font-size: smaller;
      }
   </style>
</head>

<body>
   <div class="container">
  <h1> Feedback!</h1>
<table width="100%">
   <tr>

      <p class="thank-you">Thank You for Your Feedback</p>
      <td><a href="{{url_for('feedback')}}">Submit for another response</a></td>
      <td><a href="http://127.0.0.1:5000">Go back to the homepage</a></td>
   </tr>
</table>
   </div>
</body>

</html>
```

CHAPTER -5

SCREEN SHOTS

5. SCREEN SHOTS

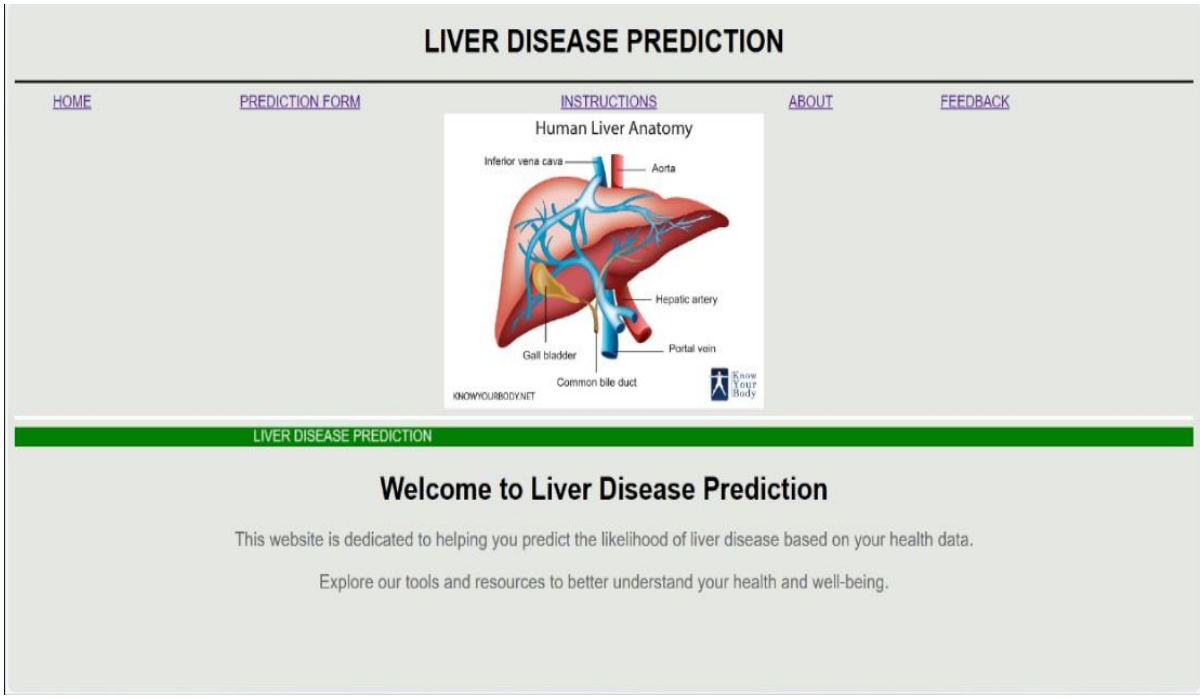


Fig: 5.1 LIVER DISEASE PREDICTION WEBPAGE

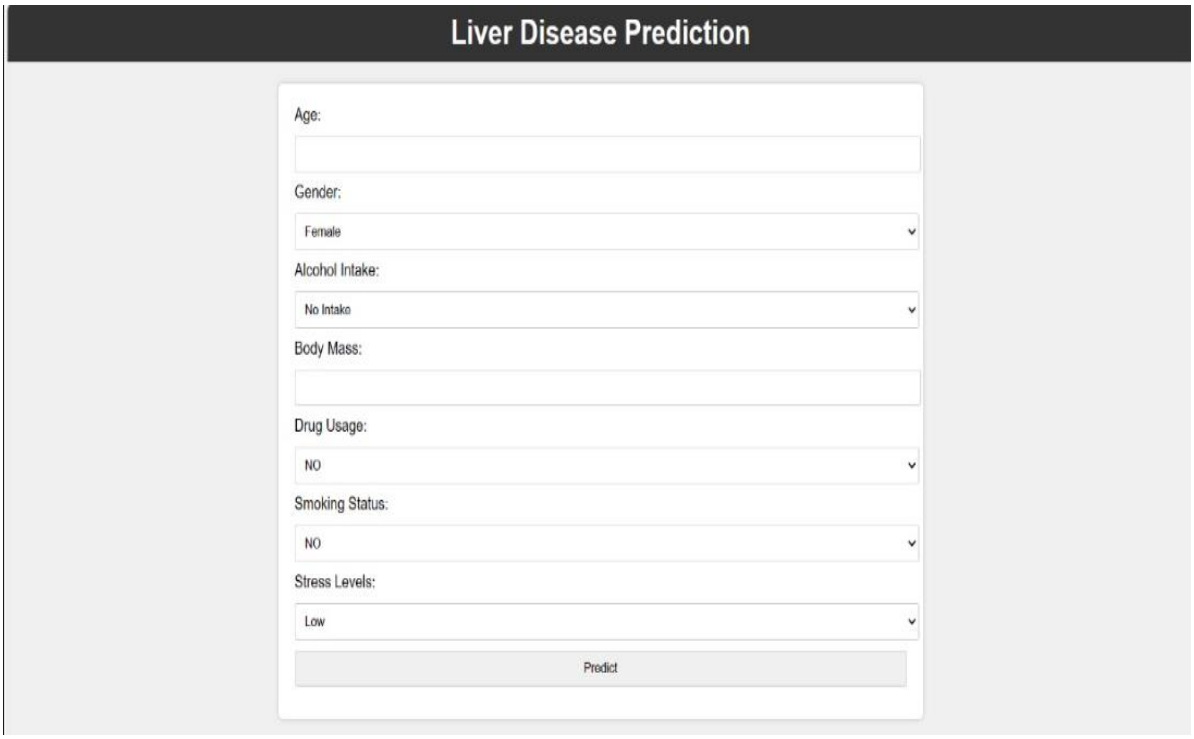


Fig: 5.2 LIVER DISEASE PREDICTION FORM

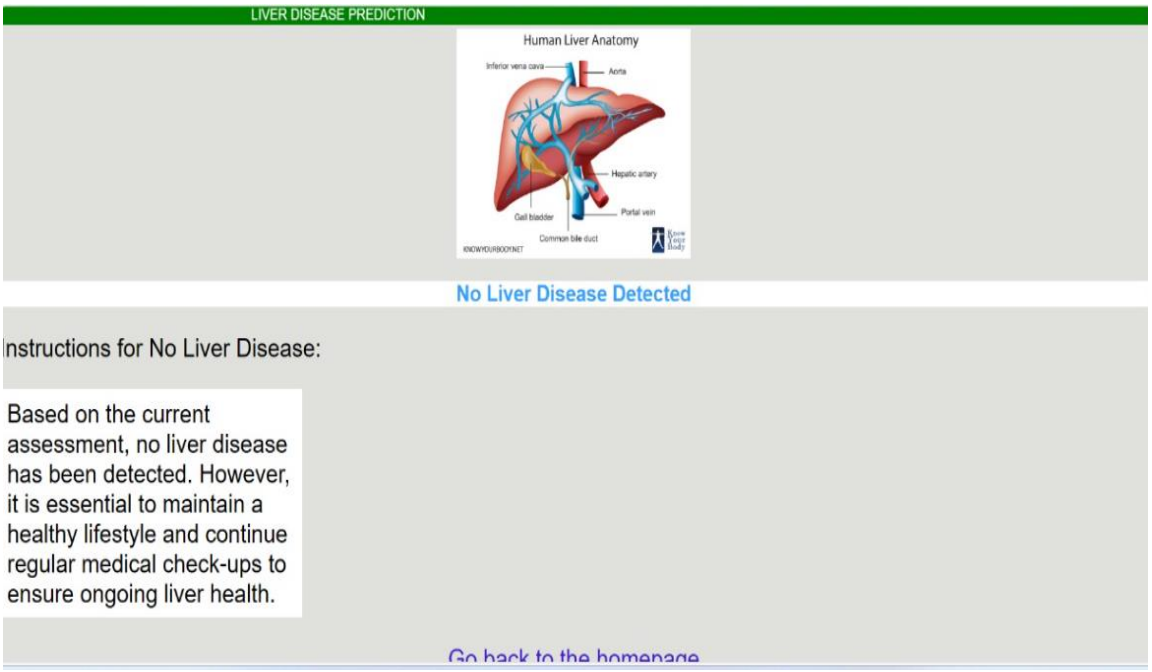


FIG 5.3: RESULT OF PREDICTION FORM

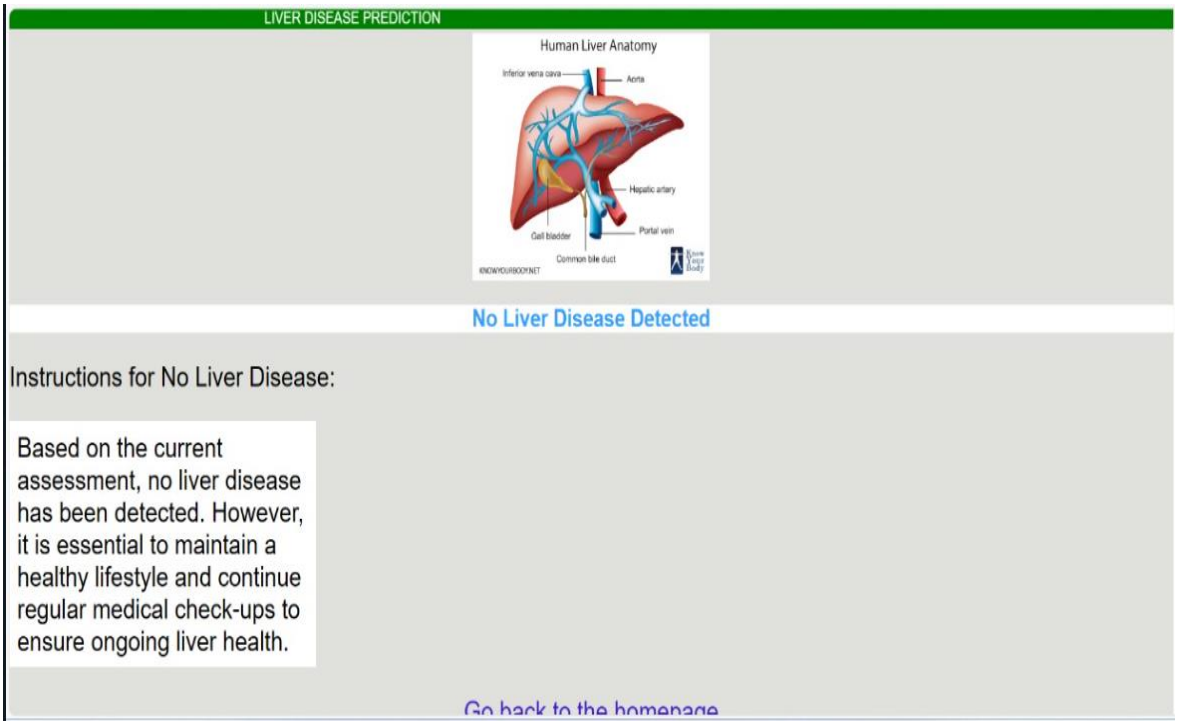


FIG 5.4: RESULT OF PREDICTION FORM

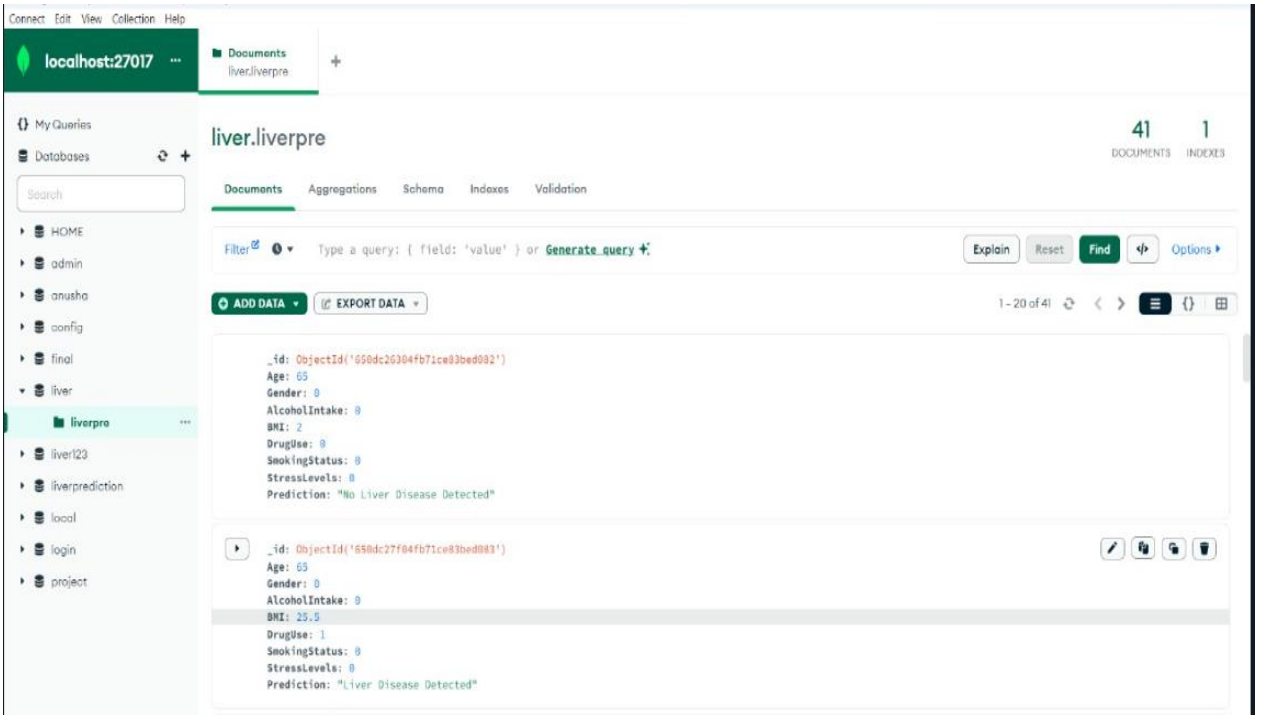


Fig: 5.5. PATIENT INPUT IS STORED IN THE DATABASE

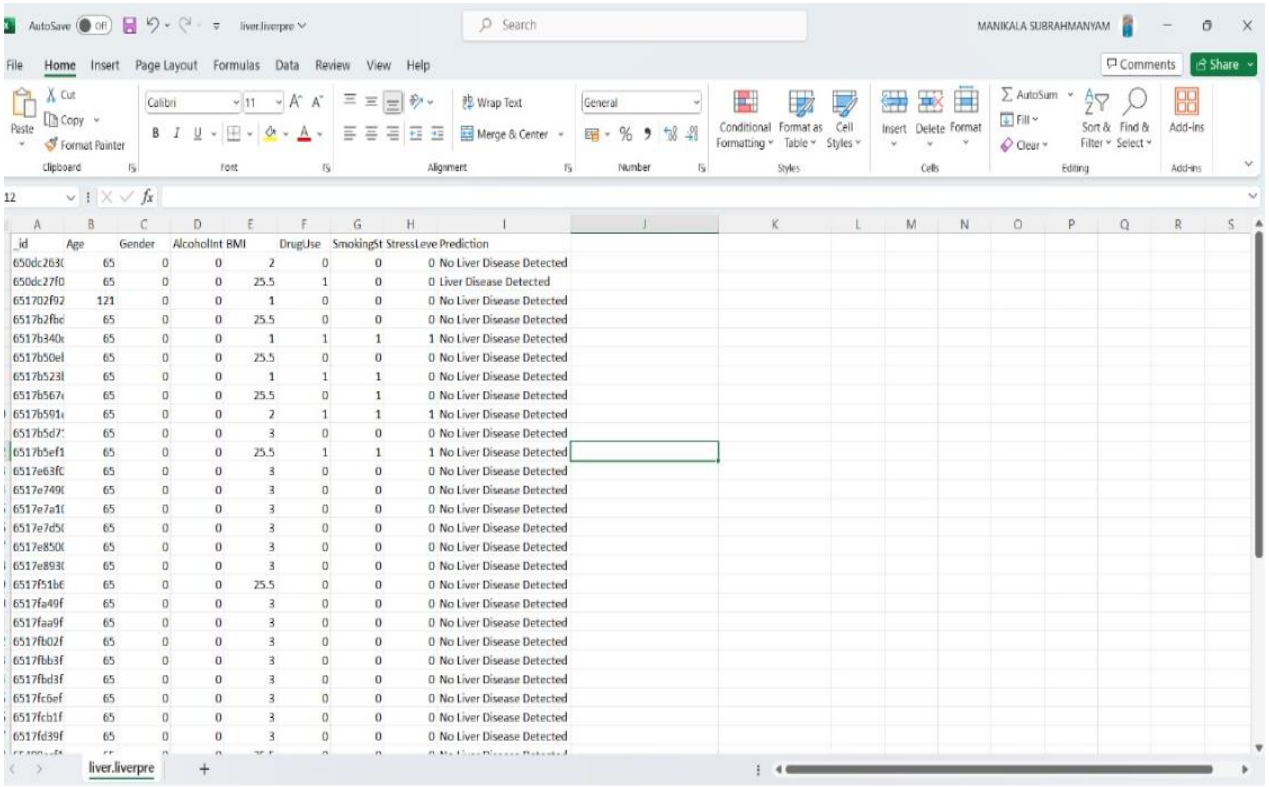


Fig: 5.6. PATIENT INPUT IN EXCEL PAGE

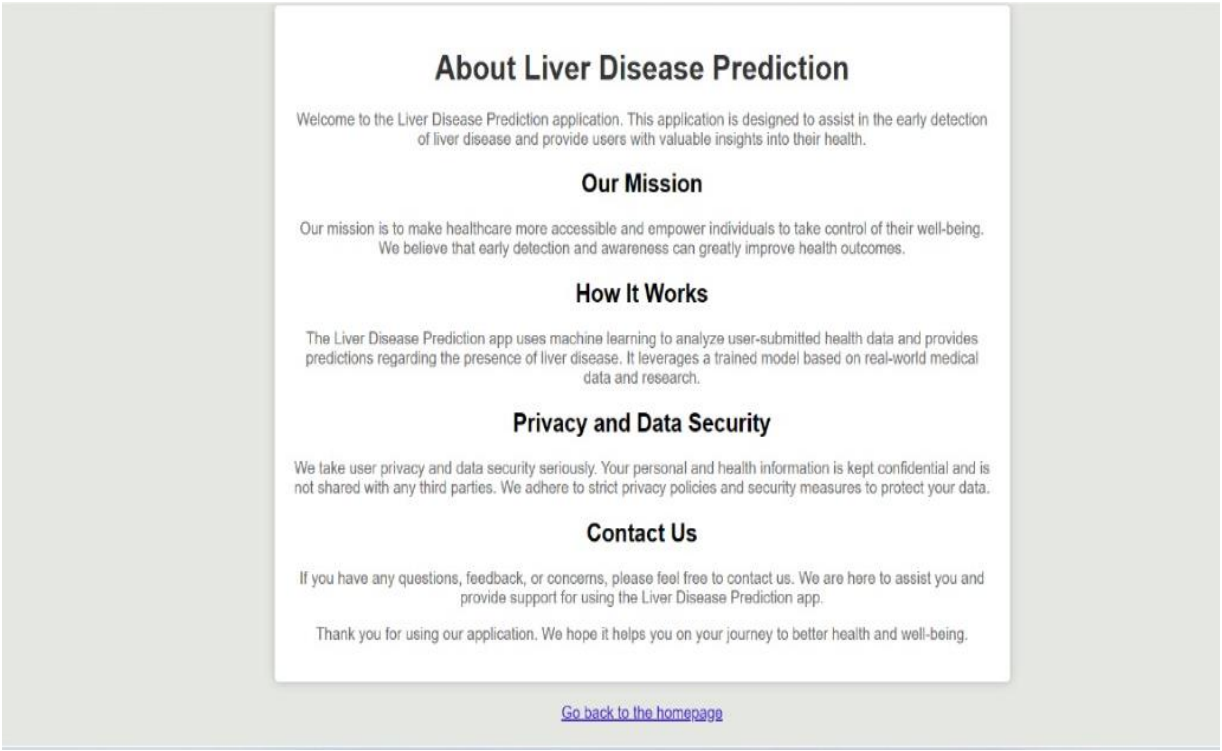


Fig: 5.7. ABOUT OUR LIVER DISEASE PREDICTION WEBPAGE

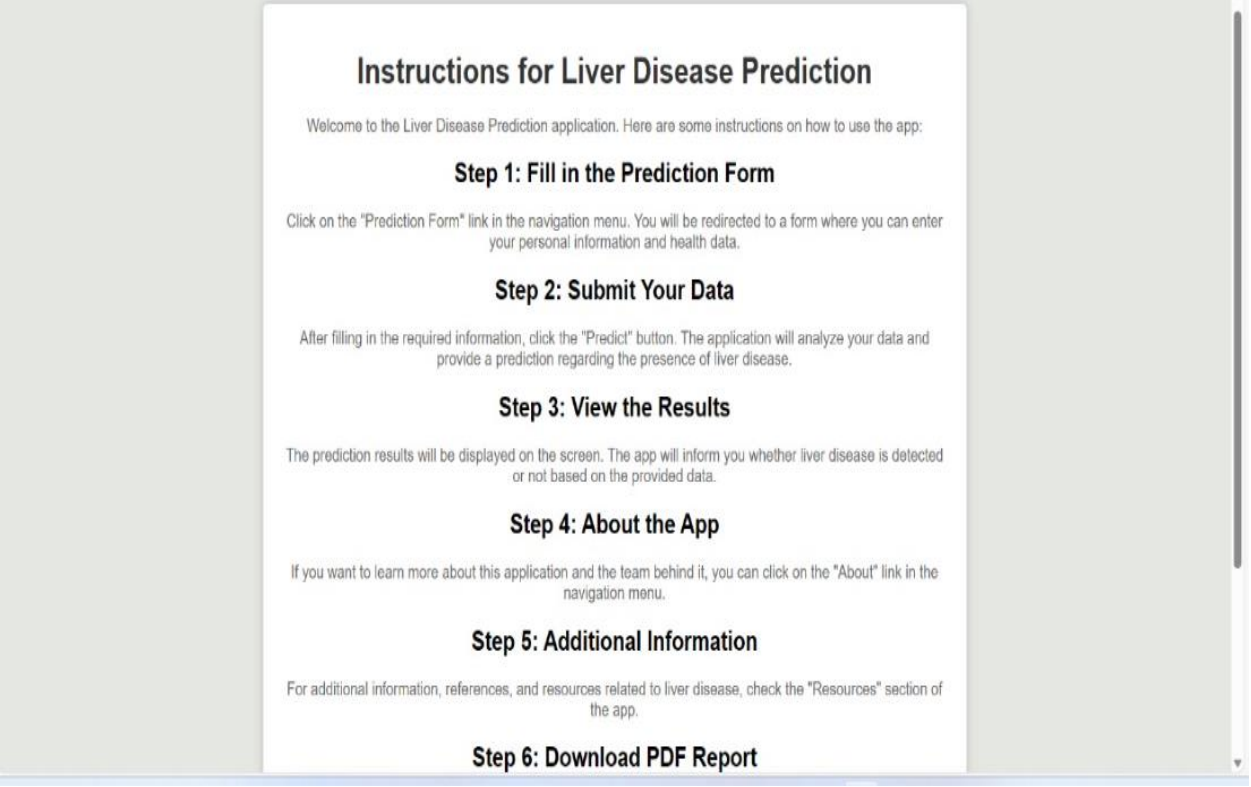


Fig: 5.8. INSTRUCTIONS FOR OUR WEBPAGE

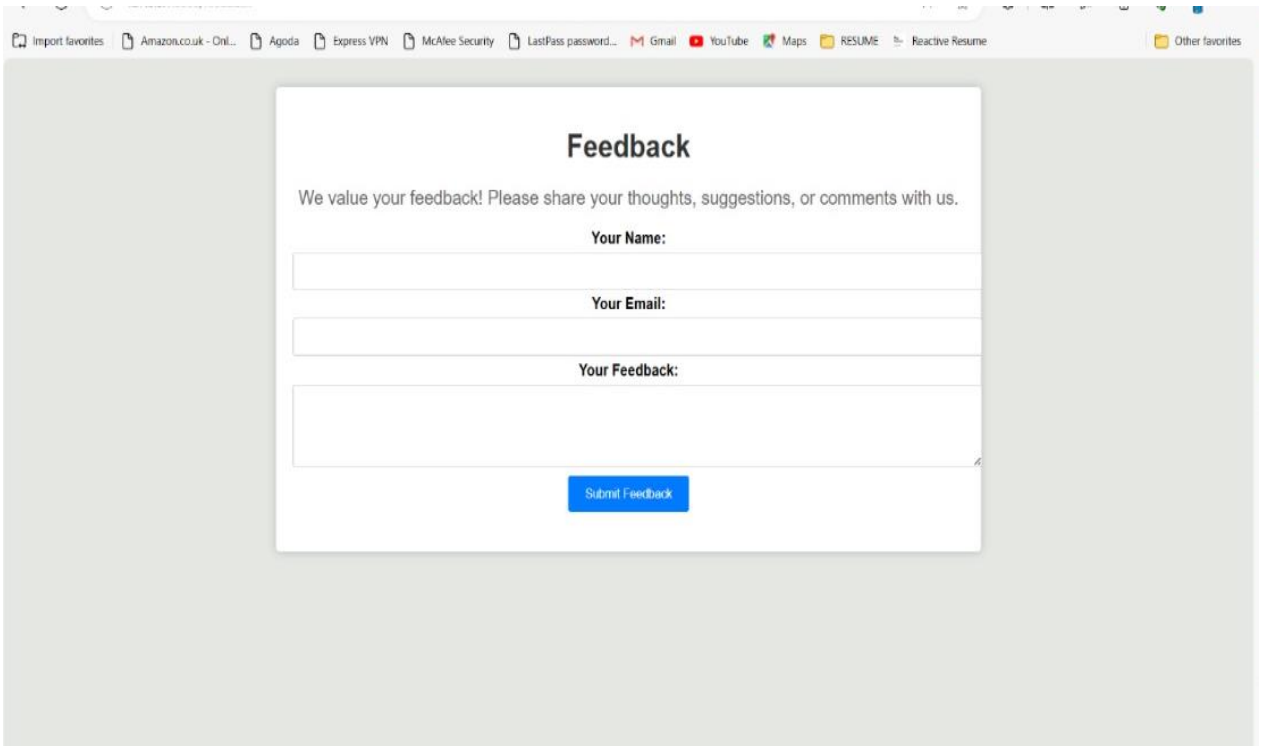


Fig: 5.9. FEED BACK FORM

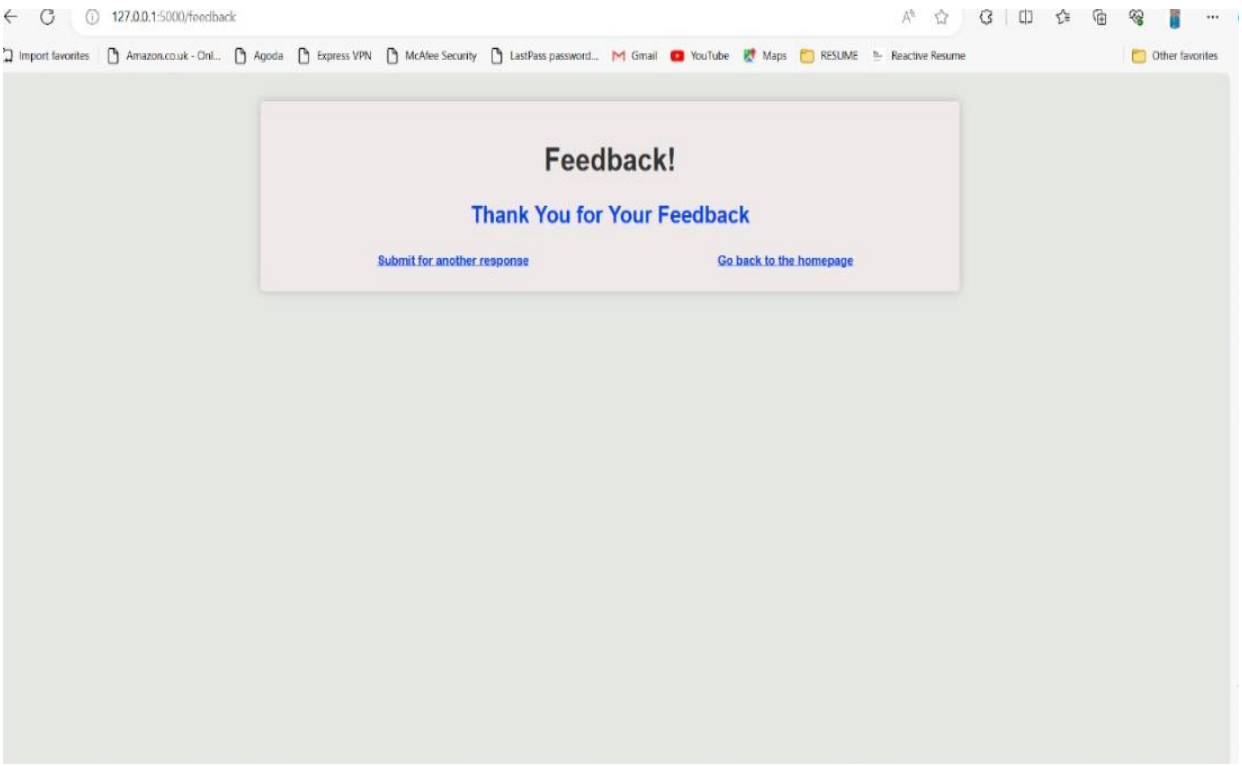


Fig: 5.10. FEEDBACK FORM

CHAPTER -6

TESTING

6. SYSTEM TESTING

6.1 TESTING STRATEGIES

TESTING

Testing is a critical phase in the development of the Liver Disease Risk Prediction System based on lifestyle attributes. It ensures that the system functions correctly and reliably, providing accurate predictions. Testing is a multifaceted process that encompasses various strategies to verify the system's functionality and evaluate its performance.

6.1.1 UNIT TESTING

Unit testing is the first line of defense in ensuring the accuracy of the system. This testing technique involves evaluating individual modules in isolation to identify and rectify defects. In the context of our project, it focuses on the functional correctness of standalone components. The primary goal is to isolate each unit of the system, analyze its behavior, and address any issues. The two main unit testing techniques employed are Black Box Testing and White Box Testing. Black Box Testing assesses the user interface, input, and output, while White Box Testing scrutinizes the behavior of each function within the system.

6.1.2 DATA FOW TESTING

Data flow testing is a specific approach within the testing phase. It involves selecting paths through the program's control flow to explore sequences of events related to the status of variables or data objects. In the context of our project, data flow testing concentrates on the points at which variables receive values and where these values are used. By exploring these data flow paths, it helps verify that data is appropriately handled, enhancing the reliability of the system.

6.1.3 INTEGRATION TESTING

After unit testing, the modules or units are integrated, giving rise to integration testing. This step aims to ensure the functional, performance, and reliability of the system concerning the interaction between the integrated modules. In our project, it verifies that various components work cohesively, validating their interconnection and compatibility.

6.1.4 BIG BANG INTEGRATION TESING

Big Bang Integration Testing is a specific integration testing strategy wherein all units are linked simultaneously, forming a complete system. While this approach accelerates integration, it makes it challenging to isolate errors found during testing, as there is limited focus on verifying the interfaces across individual units. Despite the challenges, it plays a crucial role in assessing the performance of the fully integrated system.

6.1.5 USER INTERFACE TESTING

User interface testing focuses on identifying defects in the product's graphical user interface (GUI). In our project, this technique aims to ensure the system's user interface functions as intended, providing a smooth and user-friendly experience.

The testing phase of the Liver Disease Risk Prediction System is a rigorous and systematic process, covering unit testing to evaluate individual components, data flow testing to track the journey of data within the system, integration testing to confirm the synergy of components, and user interface testing to assess the GUI's functionality. These strategies work collectively to validate the system's capabilities and ensure its reliability and accuracy.

Test Case 1 - Data Collection

Input

The user provides lifestyle attribute data, including diet, physical activity, stress levels, alcohol consumption, and smoking habits.

Output

The system collects and processes the input lifestyle data.

Result

The data collection process is successful, and the system has gathered the required lifestyle attribute data for analysis.

Test Case 2 - Feature Selection

Input

The system's feature selection module is given the lifestyle attribute data.

Output

The module selects relevant lifestyle attributes to improve prediction accuracy.

Result

The feature selection process identifies key lifestyle attributes contributing to liver disease risk, enhancing prediction accuracy.

Test Case 3 - Algorithm Testing

Input

The selected features and lifestyle data are provided as input to machine learning algorithms (e.g., Logistic Regression, SVM, KNN).

Output

The algorithms make predictions based on lifestyle attributes.

Result

The algorithms successfully predict liver disease risk using lifestyle attributes, with accuracy levels up to 79%.

Test Case 4 - Integration Testing

Input

The system combines different modules, including data collection, feature selection, and algorithm prediction.

Output

The integrated system processes lifestyle attribute data and provides predictions.

Result
The integration of modules is seamless, ensuring the system operates cohesively.

Test Case 5 - Data Flow Testing
Input
Lifestyle attribute data is tracked as it moves through the system's data processing components.
Output
The system monitors the flow of data from input to output.
Result
The data flow is accurate, and variables are appropriately handled, minimizing data-related errors.

Test Case 6 - User Interface Testing
Input
The system's graphical user interface (GUI) is accessed.
Output
The GUI is examined for usability and functionality.
Result
The GUI offers a user-friendly experience, and all functions work as intended.

Test Case 7 - User Interface Testing (Event-based)
Input
The GUI is tested with various user interactions.
Output
User interactions with the GUI are observed and assessed.
Result
The GUI handles user interactions effectively, and the system maintains its accuracy and functionality.

These test cases encompass different aspects of the system, from data collection and feature selection to algorithm testing, integration, data flow, and GUI functionality. They help ensure that the Liver Disease Risk Prediction System based on lifestyle attributes operates reliably and effectively.

CHAPTER -7 SUMMARY & CONCLUSION

7. SUMMARY & CONCLUSION

SUMMARY

The Liver Disease Risk Prediction project is a groundbreaking initiative designed to enhance the early detection and prediction of liver diseases by leveraging lifestyle attributes through binary classification. Liver diseases represent a significant global health challenge, impacting millions of lives each year. Timely diagnosis and effective interventions are vital to mitigate the impact of liver disorders. In response to this critical need, the project adopts a novel approach that incorporates lifestyle attributes alongside clinical data to create a more comprehensive model for liver disease prediction.

Clinical data, primarily focused on blood tests, has traditionally been the cornerstone of liver disease prediction. These tests assess vital parameters such as liver enzyme levels, bilirubin, albumin, and platelet counts. Machine learning algorithms have been employed to analyze this clinical data, providing insights into the presence or risk of liver disease. However, the existing system has limitations. It may not capture the complete spectrum of an individual's health, often concentrating on specific biomarkers and medical history while disregarding lifestyle factors that can significantly influence liver health. This approach can lead to missed diagnoses and incomplete risk assessments.

The proposed system challenges this status quo. By integrating lifestyle attributes such as dietary habits, physical activity, stress levels, alcohol consumption, and smoking habits, the project strives to develop a more holistic and comprehensive model for liver disease prediction. This approach not only considers clinical data but also takes into account individuals' daily choices and habits, which have a substantial impact on liver health. By applying advanced machine learning algorithms to this enriched dataset, the project aims to improve the accuracy and timeliness of liver disease prediction. This innovative approach aligns with the growing importance of preventive medicine and personalized healthcare, ultimately leading to better health outcomes and reducing the burden of liver diseases on individuals and healthcare systems.

CONCLUSION

In conclusion, the Liver Disease Risk Prediction project holds the potential to revolutionize liver disease prediction by incorporating lifestyle attributes into the existing clinical data-driven model. Lifestyle attributes are pivotal in understanding the overall health of individuals and can significantly influence the risk of liver diseases. By combining these attributes with advanced machine learning algorithms, the project addresses the limitations of the existing system.

The success of this project underscores the importance of comprehensive healthcare models that consider not only clinical data but also the daily habits and choices of individuals. This innovative approach is not only about enhancing liver disease prediction but also about contributing to a broader understanding of how lifestyle attributes can be integrated into healthcare practices. It enables early disease detection, personalized healthcare, and better health outcomes.

The Liver Disease Risk Prediction project sets the stage for future advancements in preventive medicine and individualized health management. It emphasizes the critical role of lifestyle factors in healthcare, with the ultimate goal of improving the quality of life for individuals at risk of liver diseases. This project marks a significant step towards more effective and holistic healthcare practices.

CHAPTER -8

FUTURE ENHANCEMENT

7. FUTURE ENHANCEMENT

While the Liver Disease Risk Prediction project offers a comprehensive approach to liver disease prediction based on lifestyle attributes, there are several areas for future enhancements and refinements:

Integration of Wearable Devices: In the future, the project could incorporate data from wearable health monitoring devices such as smartwatches and fitness trackers. These devices can provide real-time data on vital signs, physical activity, and sleep patterns, offering a continuous stream of lifestyle information that can further enhance prediction accuracy.

Diverse Lifestyle Factors: Expanding the range of lifestyle attributes considered could lead to more precise predictions. Factors like sleep quality, exposure to environmental toxins, and social determinants of health can play a role in liver health and should be integrated into the model.

Longitudinal Data Analysis: The project can benefit from the analysis of longitudinal data, tracking changes in lifestyle attributes over time. Long-term trends can provide insights into how specific lifestyle changes influence liver disease risk.

Personalized Recommendations: Developing a feature that offers personalized recommendations to individuals based on their lifestyle attributes can empower users to take proactive steps to improve their liver health.

Geospatial Data Integration: Considering geographic and environmental data, such as air quality and water quality in a region, can provide a more holistic understanding of liver disease risk factors.

Machine Learning Model Advancements: Continued advancements in machine learning algorithms and techniques can further enhance prediction accuracy. Techniques like deep learning and ensemble methods can be explored for better results.

Telehealth Integration: Linking the prediction system with telehealth services can provide immediate guidance and support for individuals identified as at risk, facilitating early interventions.

Population Health Studies: Collaborating with public health agencies and researchers to conduct population-level studies can help validate the effectiveness of the system on a broader scale.

Privacy and Security Measures: Enhancements in data privacy and security are crucial to protect individuals' sensitive health and lifestyle data. Implementing robust encryption and access control mechanisms is essential.

Clinical Validation: Conducting clinical trials and studies to validate the effectiveness of the lifestyle attribute-based prediction system in real-world healthcare settings.

These future enhancements aim to make the Liver Disease Risk Prediction project even more effective and comprehensive. They align with the evolving landscape of healthcare, emphasizing the importance of personalized and preventive medicine in improving health outcomes and reducing the burden of liver diseases.

CHAPTER-9

BIBLIOGRAPHY

9. BIBLIOGRAPHY

[1] In "Software-based Prediction of Liver Disease with Feature Selection and Classification Techniques" (2020) by Singh, Bagga, and Kaur, the authors focus on liver disease prediction using software engineering, classification, and feature selection methods, applying multiple classification algorithms

[2] In the research paper titled "Software-based Prediction of Liver Disease with Feature Selection and Classification Techniques" (2020) authored by Singh, Bagga, and Kaur, the primary emphasis is on the prediction of liver disease through the utilization of software engineering, classification techniques, and feature selection methods.

[3] Tanwar and Rahman's paper, "Machine Learning in liver disease diagnosis: Current progress and future opportunities," delves into the increasing use of AI-driven decision-making systems in healthcare. They discuss how these systems leverage big data and machine learning to assist in accurate disease prediction and diagnosis, particularly for liver diseases.

[4] Keerthana, Phalinkar, Mehere, Reddy, and Lal's research, "A Prediction Model of Detecting Liver Diseases in Patients using Logistic Regression of Machine Learning," focuses on addressing the significant issue of liver diseases in India. They highlight the substantial number of deaths and new diagnoses associated with liver diseases in the country.

[5] The paper "Detection of Liver Disease Using Machine Learning Approach" explores machine learning algorithms, including SVM and KNN, to predict liver disease, emphasizing early diagnosis and therapy for improved patient outcomes. The authors are Pravin Ramdas Kshirsagar, Dhoma Harshavardhan Reddy, Mallika Dhingra, Dharmesh Dhabliya, and Ankur Gupta.