**A MINI PROJECT REPORT**

**ON**

## Machine Learning Approach for Liver Disease Classification and Prediction

Submitted in the partial fulfilment of the requirements for the award of

**BACHELOR OF TECHNOLOGY**

**IN**

**ARTIFICIAL INTELLIGENCE & DATA SCIENCE**

**SUBMITTED BY**

## Name Of the Student

Naresh Bista - 21BK1A7239

G.Venu - 21BK1A7220

L.Harsha Surya - 21BK1A7223

**UNDER THE ESTEEMED GUIDANCE OF**

**Mr. Sunny Singh**

##### Assistant Professor, (Artificial Intelligence & Data Science)



**DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE**

**St. Peter’s Engineering College (UGC Autonomous)**

### Approved by AICTE, New Delhi, Accredited by NBA and NAAC with ‘A’ Grade,

**Affiliated to JNTU, Hyderabad, Telangana**

**2021-2025**



## DEPARTMENT OF

**ARTIFICIAL INTELLIGENCE & DATA SCIENCE**

**CERTIFICATE**

This is to certify that Mini Project entitled **“Machine Learning Approach for Liver Disease Classification and Prediction”,** done by Naresh Bista (21BK1A7239), G.Venu (21BK1A7220), L.Harsha Surya (21BK1A7223) in partial fulfilment for the award of the degree of **Bachelor of Technology in ARTIFICIAL INTELLIGENCE & DATA SCIENCE** is a record of bonafide work done by him/her under my supervision during the academic year “2023 – 2024”.

**INTERNAL GUIDE**

Mr. Sunny Singh

**Assistant Professor**,

Department of CSE(AI&ML)

St. Peter’s Engineering College, Hyderabad **HEAD OF THE DEPARTMENT**

Ms. M. Arsha Reddy

**Assistant Professor HOD**

Department of CSE(AI&ML)

St. Peter’s Engineering College,

Hyderabad

#### PROJECT COORDINATOR

**Mr. Chinaguravaiah Makkena**

**Associate Professor,**

Department of CSE(AI&ML)

St. Peter’s Engineering College,

Hyderabad

**EXTERNAL EXAMINER**

## ACKNOWLEDGEMENT

We sincerely express my deep sense of gratitude to **Ms. Sunny Singh** for her valuable guidance, encouragement and cooperation during all phases of the project.

We greatly indebted to my Project Coordinator M**r. Chinaguravaiah Makkena ,** for providing his valuable advice, constructive suggestions and encouragement without whom it would not been possible to complete this project.

It is a great opportunity to render my sincere thanks to **Ms. M. Arsha Reddy,** Head of the Department, CSM (AI&ML) for her timely guidance and highly interactive attitude which helped us a lot in successful execution of the Project.

We extremely thankful to our Principal **Dr. K. Sree Latha,** who stood as an inspiration behind this project and heartfelt for her endorsement and valuable suggestions.

We respect and thank our secretary, **Sri. T. V. Reddy,** for providing us an opportunity to do the project work at **St. Peter’s Engineering College** and We am extremely thankful to him for providing such a nice support and guidance which made us to complete the project.

We also acknowledge with a deep sense of reverence, my gratitude towards our parents, who have always supported me morally as well as economically. We also express gratitude to all my friends who have directly or indirectly helped me to complete this project work. We hope that I can build upon the experience and knowledge that I have gained and make a valuable contribution towards the growth of the society in coming future.

# DECLARATION

We hereby declare that the project entitled, “**Machine Learning Approach for Liver Disease Classification and Prediction** ”, is the work done during the AY 2023-24 and is submitted as project in partial fulfillment for the award of degree of Bachelor of technology in **Artifical Intelligence & Data Science**  from St. Peter’s Engineering College affiliated to JNTUH.

**Names Rollno’s**

**Naresh Bista 21BK1A7239**

**G.Venu 21BK1A7220**

**L.Harsha Surya 21BK1A7223**



## INSTITUTE VISION

To be a vibrant nodal center for Artificial Intelligence and Data Science Education, Research that make the students to contribute to technologies for IT, IT-Enabled Services; to involve in innovative research on thrust areas of data science industry and academia; to establish start-ups supporting major players in the industry.

## INSTITUTE MISSION

**DM1:** Emphasize project based learning by employing the state-of art technologies, algorithms in software development for the problems in Data science using AI.

**DM2:** Involve stakeholders to make the students industry ready with training in skill-oriented computer application software.

**DM3:** Facilitate to learn the theoretical nuances of AI techniques, Computer Engineering courses and motivate to carry out research in both core and applied areas of AI.



**DEPARTMENT OF**

**ARTIFICIAL INTELLIGENCE & DATA SCIENCE**

**DEPARTMENT VISION**

To be a vibrant nodal center for Artificial Intelligence and Data Science Education, Research that make the students to contribute to technologies for IT, IT-Enabled Services; to involve in innovative research on thrust areas of data science industry and academia; to establish start-ups supporting major players in the industry.

## DEPARTMENT MISSION

**DM1**: Create centers of excellence in cutting-edge computing and artificial intelligence

**DM2**: Impart rigorous training to generate knowledge through the state-of-the-art concepts and technologies in Artificial Intelligence and Data Science.

**DM3:** To enhance research in emerging areas by collaborating with industries and institutions at the

national and international levels.



**PROGRAM OUTCOMES (POs)**

**PO1: Engineering Knowledge**:

Apply the knowledge of mathematics, science, engineering fundamentals and provide solutions in the engineering specialization of artificial intelligence and machine learning.

**PO2: Problem Analysis:**

Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

**PO3: Design/development of solutions:**

Design solutions for complex engineering problems and design system components or processes that meet

the specified needs with appropriate consideration for the public health and safety, and the cultural,

and environmental considerations

**PO4: Conduct investigations of complex problems:**

Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

**PO5: Modern tool usage:**

Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

**PO6: The engineer and society:**

Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.



### PO7: Environment and sustainability:

Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

### PO8: Ethics:

Apply ethical principles and commit to professional ethics and responsibilities and norms of the

engineering practice.

### PO9: Individual and team work:

Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

### PO10: Communication:

Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective Presentations, and give and receive clear instructions.

### PO11: Project management and finance:

Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary Environments.

### PO12: Life-long learning:

Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.



**PROGRAM EDUCATIONAL OBJECTIVES (PEOs)**

#### PEO1:

Work effectively in interdisciplinary field with the knowledge of Artificial Intelligence and Machine Learning to develop solutions to the real-world problems.

#### PEO2:

To communicate and work effectively on team-based engineering projects and will practice the ethics of their profession consistent with a sense of social responsibility.

#### PEO3:

Excel as socially committed engineers or entrepreneurs with high ethical and moral values.

**Program Specific Outcomes:**

**PSO1:** Apply fundamental concepts of Artificial Intelligence and Machine Learning to solve multidisciplinary engineering problems.

**PSO2:** To communicate and work effectively on team based engineering projects and will practice the

ethics of their profession consistent with a sense of social responsibility.

# TABLE OF CONTENTS

**ABSTRACT III**

**LIST OF FIGURES V**

**LIST OF SCREENSHOTS IV**

#### INTRODUCTION 1-3

* 1. Motivation
  2. Problem Definition
  3. Objective of the Project
  4. Limitations

#### LITERATURE SURVEY 4-6

* 1. Existing System
  2. Disadvantages of Existing System
  3. Proposed System
  4. Advantages of Proposed System

#### ANALYSIS 7-12

* 1. Requirement Specification
  2. Software Requirements

#### DESIGN 13-15

* 1. System Architecture
  2. Input design
  3. Output design

## IMPLEMENTATION 16-22

* 1. Technologies used
  2. Sample code
  3. Flow charts

## TESTING AND VALIDATION 23-27

* 1. Introduction
  2. Test Approach
  3. Validation

1. **RESULTS 28-30**
2. **CONCLUSION AND FUTURE SCOPE 31**
3. **REFERENCES 32**

**ABSTRACT**

Liver disease is a significant global health issue affecting millions worldwide. Early detection

and classification of liver diseases are crucial for timely intervention and effective treatment.

Machine learning (ML) techniques have emerged as powerful tools in medical diagnostics due to their

ability to analyze large datasets and identify complex patterns. In this study, we propose a machine

learning approach for liver disease classification and detection using a dataset with information about

patients, demographic including their age, gender, and liver function test results. In this project we will

be using different machine learning methods, such as CatBoost(Categorical Boosting), XGBoost

(Extreme Gradient Boosting), and LightGBM(Light Gradient Boosting) to find the best model.

We evaluate how well these models work using measures like accuracy. The best model is turned into

an easy-to-use web application for healthcare professionals. This project shows that machine learning

can help in the early detection of liver disease, providing a useful tool for doctors to screen patients

and start treatment early.

We will be using Google Colab IDE for implementing our project. Google Colab comes pre-installed with many popular Python libraries such as NumPy, Pandas, Matplotlib, and Scikit-learn, making it easy to start coding without worrying about installation issues. its support interactive data analysis and visualization Libraries such as Matplotlib and Seaborn are commonly used for creating visualizations that help in understanding data distributions, correlations, and model performance metrics. Implementing a project for liver disease classification and detection using Google Colab involves several key steps; Healthcare , Data Collection , Data Cleaning and Preprocessing, Exploratory Data Analysis (EDA), Feature engineering and selection, Model selection and training Model evaluation and validation.

## INDEX

|  |  |  |
| --- | --- | --- |
| **S.NO** | **PARTICULARS** | **PAGENO** |
| 1 | INTRODUCTION | 1 |
| 2 | LITERATURE SURVEY | 4 |
| 3 | ANALYSIS | 7 |

1. DESIGN 13
2. IMPLEMENTATION 16

1. TESTING AND VALIDATION 23

7 RESULTS 28

8 CONCLUSION AND FUTURE SCOPE 31

9 REFERENCES 32

**LIST OF FIGURES**

**S.NO PARTICULARS PAGE NO**

|  |  |  |
| --- | --- | --- |
| 1  2      3    4 | Architecture of the Liver  Disease prediction  The Sequence Diagram for the System Development  Sample output for Execution  Output in an execution  environment | 13  22  28  30 |

**LIST OF SCREENSHOTS**

**S.NO PARTICULARS PAGE NO**

|  |  |  |
| --- | --- | --- |
| 1  2      3 | Diagnosis Distribution  Predict Liver Disease Frontend Interface  Output Report | 37  38  39 |

#### 1. INTRODUCTION

Liver disease is a major global health concern, causing significant illness and death each year.

The Liver a vital organ, performs essential functions such as detoxification, protein synthesis, and the production of biochemicals necessary for digestion. When the liver is compromised, it can lead to

Severe health problems, including liver cirrhosis, hepatitis, and liver cancer. Early detection and

Diagnosis of liver disease are crucial for effective treatment and management. Traditional diagnostic methods often involve invasive procedures and can be time-consuming. With the advancement of data science and machine learning, there is a growing potential to develop non-invasive, accurate, and

Efficient tools for the early detection and classification of liver disease.

This project, titled "Machine Learning Approach for Liver Disease Classification and

Prediction," aims to harness the power of machine learning to create a predictive model that can

accurately identify individuals at risk of liver disease. By analyzing a comprehensive dataset

containing patient demographics, lifestyle factors, and clinical test results, the project seeks to build

a robust model that can assist healthcare professionals in early diagnosis and intervention.

In recent years, the application of machine learning techniques in medical diagnostics has

Shown promising results, particularly in the realm of liver disease detection and classification.

Liver diseases, encompassing a wide range of conditions from hepatitis to cirrhosis, pose

significant health challenges globally. Early and accurate diagnosis is crucial for effective treatment

and management of these conditions. Through this project, we aim to contribute to the advancement

of medical technology by providing a tool that enhances diagnostic accuracy and supports

healthcare providers in delivering timely interventions for patients with liver diseases.

Key objectives of this project include are;

## 1.MOTIVATION

The motivation behind the project "Machine Learning Approach for Liver Disease Classification and Prediction" lies in leveraging the power of machine learning to enhance liver disease diagnosis, improve patient outcomes, optimize healthcare resources, and contribute to scientific progress in medical technology. These factors collectively drive the pursuit of developing effective and efficient tools for liver disease Prediction and classification.

## 1.2.PROBLEM DEFINITION

The project aims to address the challenge of accurate and timely diagnosis of liver diseases

using machine learning techniques. Liver diseases encompass a diverse range of conditions with

varying symptoms and underlying causes, making diagnosis complex and often requiring

specialized expertise. Current diagnostic methods may not always detect diseases in their early

stages, leading to delayed treatment and poorer patient outcomes. The primary problem is to

develop a machine learning model capable of accurately classifying different types of liver

diseases based on diverse sets of medical data, including clinical indicators, patient demographics,

and diagnostic test results. This model must handle the complexity of liver disease patterns, account

for varying disease progression stages, and integrate seamlessly into clinical practice. Key

objectives include optimizing diagnostic accuracy, enhancing efficiency in healthcare delivery,

and providing healthcare professionals with a reliable tool for early detection and classification

of liver diseases. By addressing these challenges, the project aims to improve patient care

outcomes and contribute to advancing the field of medical diagnostics through AI-driven solutions.

This problem definition outlines the specific challenges the project aims to tackle, the scope of

the machine learning application, and the expected outcomes in terms of healthcare

improvement and technological advancement.Top of Form

## OBJECTIVE OF THE PROJECT

* Develop accurate machine learning models for classifying various types of liver diseases based on clinical data, patient demographics, and diagnostic test results.
* Enhance early detection capabilities to identify liver diseases in their early stages, improving treatment outcomes and patient prognosis.
* Optimize diagnostic efficiency by automating and streamlining the analysis of complex medical data using machine learning algorithms.
* Explore personalized medicine opportunities by identifying unique patterns and correlations within patient data to inform tailored treatment approaches.
* Validate and generalize the developed models to ensure reliability and applicability across diverse patient populations and healthcare environments.
* Integrate machine learning models into clinical practice through user-friendly interfaces or tools that support healthcare professionals in decision-making.
* Contribute to the advancement of medical knowledge by uncovering new insights into liver diseases through data-driven approaches.

.

## LIMITATIONS

* **Data Availability and Quality**: Limited availability or quality of comprehensive datasets containing diverse patient demographics, detailed clinical histories, and diagnostic test results could impact the model's performance and generalizability.

 **Bias in Data**: Biases inherent in the data collection process (e.g., demographic bias, selection bias)

may lead to skewed model predictions, affecting the fairness and reliability of diagnostic

outcomes, especially across diverse patient populations.

## LITERATURE SURVEY

Liver diseases represent a significant global health challenge, affecting millions of individuals

worldwide. Conditions such as hepatitis, cirrhosis, and fatty liver disease can lead to severe

health complications and, in many cases, result in death if not diagnosed and treated promptly.

Traditional diagnostic methods, while effective, often involve invasive procedures and are

time-consuming. This is where machine learning (ML) can play a transformative role. By leveraging

vast amounts of medical data, ML algorithms can aid in the early detection and classification of

liver diseases, potentially leading to better patient outcomes and more efficient healthcare systems.

Machine learning has made substantial inroads into various healthcare domains, from predictive

modeling to diagnostic support systems. In the context of liver diseases, ML techniques offer

promising solutions for enhancing diagnostic accuracy and enabling personalized treatment plans.

By analyzing patterns within complex datasets, ML can identify subtle indicators of disease that may

be missed by conventional methods. A crucial step in developing ML models for liver disease

detection involves the collection and preprocessing of data. Common sources of data include medical

records, laboratory results, and imaging data such as ultrasounds and MRIs. Preprocessing these

datasets is essential to ensure the accuracy and reliability of the ML models. Techniques such as

handling missing data, feature selection, and normalization are employed to prepare the data for analysis

## EXISTING SYSTEM

Existing System for Liver Disease Classification and Prediction the traditional approaches to diagnosing

Liver diseases rely heavily on clinical evaluations

1. **Clinical Evaluation:** Physicians perform a thorough examination of the patient's medical history, symptoms, and physical signs such as jaundice, abdominal swelling, and liver enlargement.
2. **Predictive Analytics**: Supervised learning models, specifically **CatBoost, XGBoost, and LightGBM**, are employed to predict the likelihood of liver disease based on patient data, including blood test results and clinical history. These models are effective in identifying complex patterns and risk factors that may not be immediately apparent to human clinicians, enhancing early detection and informed clinical decision-making.
3. **Automated Diagnosis**: ML systems are being developed to provide automated, real-time diagnostic support. For instance, decision support systems can analyze patient data and suggest potential diagnoses, helping clinicians make more informed decisions.
4. **Big Data Utilization**: With the vast amount of healthcare data available, ML models can analyze and learn from large datasets, leading to continuous improvements in diagnostic accuracy and the ability to identify rare liver conditions

## 2.2.DISADVANTAGES OF EXISTING SYSTEM

* Medical datasets often have missing values and inconsistencies, affecting ML model performance.
* Biases in training data can lead to biased ML models, reducing accuracy for certain patient groups.
* Advanced ML models, especially deep learning algorithms, can be complex and difficult to understand.
* Clinicians may hesitate to rely on ML models without clear explanations for their decisions.
* Integrating ML systems into existing clinical workflows can be disruptive.
* The user interface and experience of ML tools need to be intuitive for clinicians to adopt them seamlessly.
* Ensuring ML models comply with healthcare regulations and standards is complex and time-consuming.
* Protecting patient data privacy is crucial, requiring strict data security protocols.
* Continuous monitoring and updating of ML models are necessary to maintain accuracy and

relevance as new data and medical knowledge emerge.

## PROPOSED SYSTEM

The proposed system leverages advanced machine learning techniques to enhance liver

disease classification and detection. By addressing the limitations of traditional methods and

integrating seamlessly into clinical workflows, the system aims to improve diagnostic accuracy,

efficiency, and accessibility. Ensuring ethical and regulatory compliance, continuous monitoring,

and updating of models will be crucial for the successful implementation and acceptance of this

system in clinical practice.

## ADVANTAGES OF PROPOSED SYSTEM

The proposed system for liver disease classification and detection using machine learning offers

numerous advantages, including enhanced diagnostic accuracy, early detection, non-invasive

diagnostics, efficiency, accessibility, and seamless integration with clinical workflows. By ensuring

model interpretability, ethical compliance, and continuous improvement, the system aims to

transform liver disease diagnostics, ultimately leading to better patient care and outcomes.

. **Enhanced Diagnostic Accuracy:**

* **Pattern Recognition**: ML models can identify complex patterns in data that may be missed

by human clinicians, leading to more accurate diagnoses.

* **Consistency**: ML algorithms provide consistent results, reducing the variability associated

with human interpretation.

**2. Early Detection:**

* **Predictive Capabilities**: ML models can detect early signs of liver disease, enabling

timely interventions and improving patient outcomes.

* **Risk Stratification**: The system can identify high-risk patients for closer monitoring and

proactive treatment.

**3. Efficiency and Speed:**

* **Automated Analysis**: ML models can quickly analyze large volumes of data, providing

rapid diagnostic results.

* **Real-Time Support**: The system offers real-time diagnostic support, enabling faster

clinical decision-making.

**4.Accessibility and Scalability:**

* **Cost-Effective**: Non-invasive and automated diagnostic methods are generally more cost-

effective , making them accessible to a broader population.

* **Widespread Availability**: The system can be deployed in various healthcare settings,

## ANALYSIS

#### REQUIREMENT SPECIFICATION

#### 1.Project Overview:

The project aims to develop a machine learning model for accurate classification and detection

of liver diseases using clinical data. This will assist healthcare providers in early diagnosis

and treatment planning.

#### 2.Scope:

* Collecting and preprocessing clinical data related to liver diseases.
* Developing and evaluating machine learning models for classification.
* Deploying the model in a healthcare setting for practical use.

#### 3.Data Collection:

* Obtain a comprehensive dataset including patient demographics, medical history, and diagnostic

test results from reliable sources (e.g., medical institutions, public repositories).

* Ensure data includes sufficient samples representing various types and stages of liver diseases.

#### 4.Data Preprocessing:

* Handle missing data through appropriate techniques (e.g., imputation, deletion).
* Normalize numerical features to a standard scale.
* Encode categorical variables for model compatibility (e.g., one-hot encoding).

#### 5.Model Development:

* Explore and select suitable machine learning algorithms for disease classification.
* Implement feature selection techniques to identify relevant predictors.

Develop an ensemble model to enhance predictive accuracy if beneficial

#### 6 .Model Evaluation:

* Split data into training, validation, and test sets.
* Evaluate models using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.
* Conduct cross-validation to ensure model robustness and generalizability.

### 7.Data Requirements:

### 

#### Data Sources:

* Obtain data legally and ethically from healthcare providers or public datasets with

appropriate permissions.

#### Data Quality:

* Ensure data is accurate, complete, and representative of diverse patient demographics and

disease types.

#### Data Privacy:

* Implement measures to anonymize patient data during model training and inference.
* Comply with regulations regarding the handling of sensitive

medical information.

### 8.Constraints:

#### Technological Constraints:

* Use software and hardware compatible with the healthcare institution’s IT infrastructure.

#### Regulatory Constraints:

* Adhere to medical ethics guidelines and regulatory requirements governing the use of patient

data.

### 9.Assumptions and Dependencies:

#### Assumptions:

* Assumed availability of necessary computational resources for model training and deployment.
* Assumed cooperation from healthcare providers for data access and validation.

#### Dependencies:

### Dependence on the availability and quality of clinical data from authorized sources

### 10.Acceptance Criteria:

* + - Obtain approval from medical professionals for the practical use of the model in clinical

settings

* Achieve a minimum accuracy of 90% on the test dataset for disease classification.
* Successfully integrate the model into the healthcare system without disrupting existing

operations.

### 11.Risks:

* Potential risks include model bias due to unrepresentative training data.
* Risks associated with data breaches or privacy violations.
* Risks related to regulatory non-compliance affecting project timelines and deliverables

This requirement specification document outlines the key aspects necessary for successfully

developing and deploying a machine learning approach for liver disease classification and

detection. Adjustments can be made based on specific project constraints and stakeholder requirements

## SOFTWARE REQUIREMENTS

##### programming Language Used:

##### Python

##### Python libraries:

##### Pandas

##### NumPy

##### Machine learning Libraries:

##### CatBoost, XGBoost, LightGBM

##### Visualization Libraries:

##### Matplotlib, Seaborn

##### Web Development stack:

##### Flask, HTML/CSS

##### IDE’s Used:

##### Google Colab, VS Code

## METHODS AND TECHNOLOGIES INVOLVED

Developing a Machine Learning -based solution for the Detection and Prediction of Polycystic

Ovary Syndrome (LIVER). To achieve this goal, several methods and technologies would be

involved.

### Data Collection and Preprocessing:

* **Python**: For scripting and coding, including data handling and model training.
* **Pandas**: For data manipulation and analysis.
* **NumPy**: For numerical computations.
* **Scikit-Learn**: For preprocessing, including feature scaling, encoding, and splitting data.

**2. Data Cleaning and Transformation**

* **Python Libraries**: Functions within **Pandas** and **Scikit-Learn** for missing data handling
* and transformations.

**3. Machine Learning Model Development**

* **Scikit-Learn**: For baseline models and decision tree algorithms.
* **CatBoost, XGBoost, and LightGBM**: Advanced gradient-boosting algorithms for enhanced

predictive performance.

* **Matplotlib & Seaborn**: For exploratory data analysis (EDA) through data visualization.

**Supervised Learning**: Utilize labeled data to train models. Common algorithms include:

* + CatBoost (Categorical Boosting)
  + XGBoost (Extreme Gradient Boosting)
  + LightGBM (Light Gradient Boosting Machine)
* **Categorical Boosting**:
* **Definition**: CatBoost or *Categorical*, is a gradient boosting algorithm that efficiently handles categorical data within decision tree models by directly encoding categorical features without extensive preprocessing.
* **Purpose**: The purpose of CatBoost is to improve prediction accuracy and speed, particularly in datasets with many categorical variables, making it ideal for applications in fields like finance, e-commerce, and healthcare
* **Extreme Gradient Boosting**:
* **Definition**: Decision Trees are hierarchical tree-like structures where internal nodes represent feature

tests, branches represent decision outcomes, and leaf nodes represent class labels regression trees).

* **Purpose**: Decision Trees are intuitive and easy to interpret, making them useful in medical

decision-making processes. They can help identify important features for diagnosing liver disease

and can handle both categorical and numerical data effectively.

* **Light Gradient Boosting Machine:**
* **Definition:** XGBoost, short for *eXtreme Gradient Boosting*, is a high-performance, scalable

machine learning algorithm based on gradient boosting. It is designed to optimize both

computational speed and predictive accuracy in decision tree models.

1. **Deployment & Backend Development**

* **Flask**: For creating the web application to host the liver disease prediction model.
* **HTML & CSS**: For front-end structuring and styling of the web application

### 

## 

**5.** **Visualization and Interpretation**

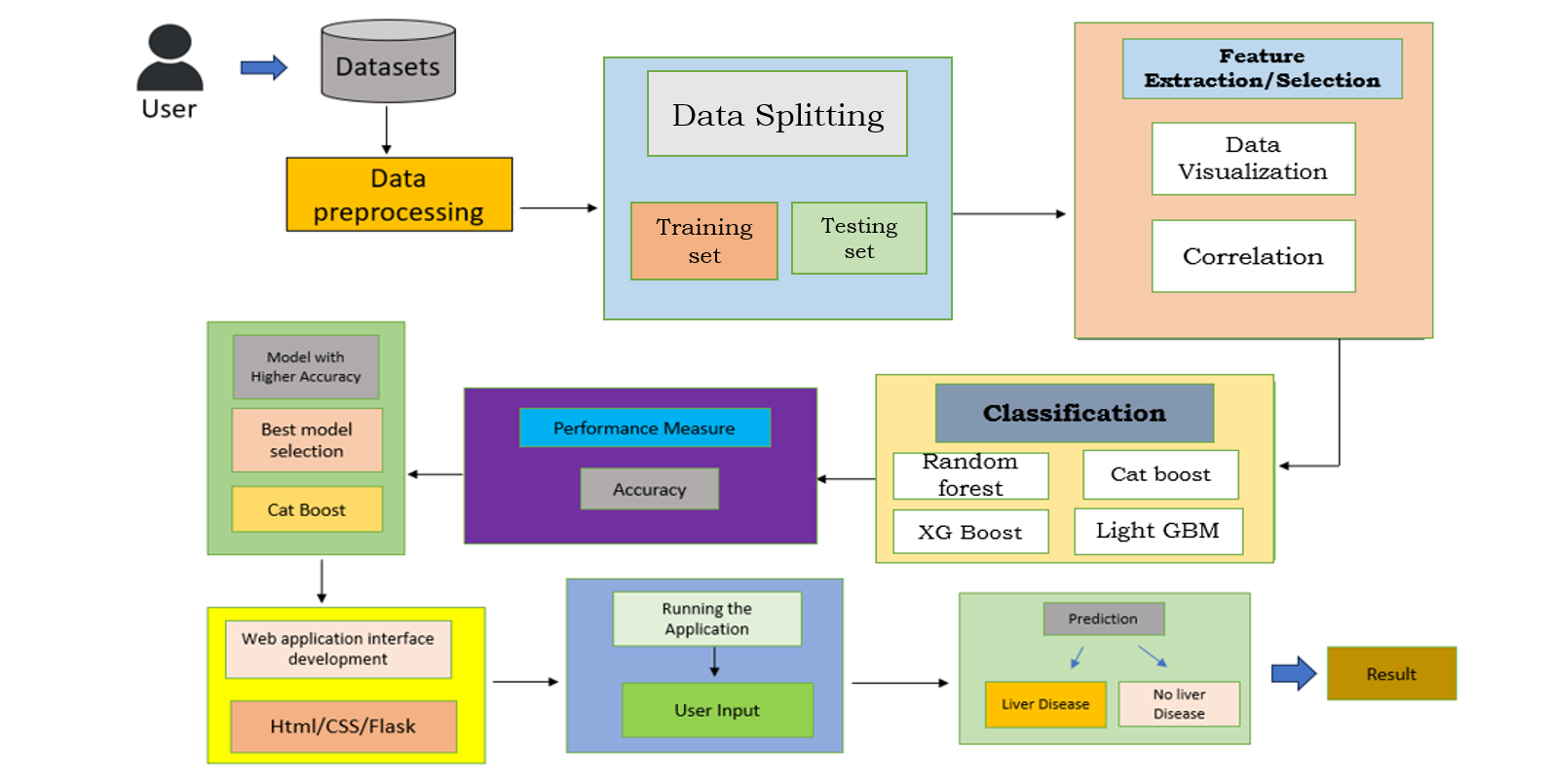
* **Matplotlib & Seaborn**: For model performance visualization and EDA charts.

**6.IDE’S USED**

* Google Colab
* VS Code

## 4.DESIGN

* 1. **SYSTEM ARCHITECTURE**

****

## 

**Data Collection:** This section represents the data sources, such as patient data and medical records, which are essential for training and evaluation.

**Data Preprocessing:** Data preprocessing steps involve cleaning and preparing the data for machine learning. This includes data cleaning, feature extraction, and data transformation.

**Machine Learning:** This section includes model training and model evaluation components. Machine learning models are trained on the preprocessed data, and their performance is evaluated.

**Prediction:** The prediction engine utilizes the trained machine learning models to make predictions about liver disease. we begin our study in this part with the data-processing stage and go on to feature extraction, classification, and prediction analysis.

## 4.2.INPUT DESIGN

Designing the input for a Machine Learning (ML)approach for liver disease classification and detection involves collecting diverse medical data (demographics, medical history, lab results) and preprocessing it (cleaning, normalization, feature engineering). Inputs are structured (tabular for numerical data, tokenized for text, preprocessed for images), split into training, validation, and test sets for model development, tuning, and evaluation. Considerations include handling data imbalance, ensuring privacy compliance, and prioritizing interpretability for clinical use, ensuring robust and effective disease classification. The quality and comprehensiveness of input data play a crucial role in the accuracy and effectiveness detection and prediction algorithms.

## 4.3 OUTPUT DESIGN

The output of a Machine Learning project focused on liver disease classification provides

actionable predictions and insights to support clinical decision-making.

* **Predictions**: Determines whether a patient has liver disease or not, aiding in initial screening

and diagnosis.

* **Classification**: Identifies specific types or stages of liver disease (e.g., fatty liver, hepatitis, cirrhosis), guiding targeted treatment strategies.
* **Probability Scores**: Quantifies the model's confidence in predictions, helping clinicians assess reliability.
* **Feature Importance**: Highlights key factors influencing predictions (e.g., liver enzymes, demographics), enhancing understanding of diagnostic markers.
* **Visualizations**: Uses charts and graphs to present results intuitively (e.g., confusion matrices, ROC curves), supporting interpretation.
* **Integration**: Facilitates seamless integration into clinical workflows via APIs or interfaces for real-time decision support.
* **Clinical Application**: Supports accurate diagnosis, personalized treatment plans, and proactive patient management.
* **Ethical Considerations**: Ensures fairness, privacy compliance, and unbiased predictions across diverse patient populations.

The output design for machine learning approach for liver disease classification and detection and prediction includes clinical reports for healthcare providers, patient dashboards, data visualization for easier comprehension, real-time alerts, predictive models for risk assessment, educational materials, and a strong focus on privacy and data security. These outputs aim to empower both healthcare professionals and patients with valuable information for informed decision

## 5. IMPLEMENTATION

### Technologies Used:

##### Python:

Python is of paramount importance in this project on "Machine learning approach for

Liver disease classification and detection" due to its exceptional capabilities in the fields of

data analysis, machine learning, and scientific computing. Python offers an extensive ecosystem

of libraries and frameworks, such as scikit-learn, matplotlib , numpy and pandas, specifically

designed for tasks crucial to this project. Its user-friendly syntax facilitates rapid development

and experimentation, crucial for fine-tuning machine learning models and exploring complex

datasets. Python's wide adoption in the machine learning community ensures a wealth of

resources, tutorials, and community support, streamlining problem- solving and

collaboration. Moreover, Python's versatility allows for seamless integration with web

development tools, databases, and visualization libraries, ensuring a holistic approach to

data-driven healthcare solutions. In sum, Python's comprehensive toolkit and strong community

make it an indispensable asset, empowering this project to effectively address the intricate

challenges of detection and prediction.

### Machine Learning:

The integration of Machine Learning (ML) into this project on "Machine learning approach for liver disease classification and detection" holds paramount importance. ML empowers the project with the ability to analyze complex and multi-dimensional patient data, enabling the creation of highly accurate predictive models for liver disease detection. Its capacity to recognize subtle patterns and relationships with in data goes beyond human capabilities, allowing for the identification of early detection of liver diseases ,in the project we are using 4 machine learning algorithms such as logistic regression, K-nearest neighbors, and decision tree.

### Google Colab:

Google Colab, a cloud-based Python development environment, plays a pivotal role in this project by offering several critical advantages. Firstly, it provides a powerful computing infrastructure, including GPUs and TPUs, allowing for the rapid training of complex machine learning models on extensive patient datasets without the need for high-end local hardware. Second, Colab offers seamless collaboration features, facilitating team collaboration and version control for code and data. Moreover, it eliminates the hassle of setting up and maintaining a local development environment, saving time and resources. Lastly, Colab's integration with Google Drive simplifies data storage and access, ensuring that project data and code are securely managed in the cloud. In summary, Google Colab greatly streamlines the development and execution of the project, making it an indispensable tool for efficient and collaborative machine learning and data analytics in the context of LIVER detection and prediction

**Flask**

Flask is a lightweight, flexible web framework for Python, designed to build web applications quickly and efficiently. Known for its simplicity, Flask provides essential features such as routing, request handling, and template rendering, allowing developers to create web servers that process user requests and display dynamic web pages. Unlike heavier frameworks, Flask doesn’t enforce specific tools or libraries, which makes it highly customizable and ideal for small to medium-sized applications or prototypes.

**HTML,CSS**

HTML (HyperText Markup Language) and CSS (Cascading Style Sheets) are foundational technologies for building web pages. HTML provides the structure and content, while CSS adds styling to make the content visually appealing**.**

### Source Code:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import plotly.express as px

!pip install catboost

!pip install xgboost

!pip install lightgbm

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler, LabelEncoder

from sklearn.metrics import accuracy\_score

from sklearn.ensemble import RandomForestClassifier

from catboost import CatBoostClassifier

from xgboost import XGBClassifier

from lightgbm import LGBMClassifier

warnings.filterwarnings('ignore')

df =pd.read\_csv('/content/Liver\_disease\_data.csv')

df.head()

df.shape

df.info()

df.describe()

df.isnull().sum()

df['Age'].value\_counts().reset\_index()

plt = px.histogram(df, x="Age", title="Age Distribution", color\_discrete\_sequence=px.colors.sequential.Cividis)

plt.show()

plt = px.pie(df, names="Gender", title="Gender Distribution", color\_discrete\_sequence=px.colors.sequential.Mint\_r)

plt.show()

plt = px.histogram(df, x="BMI", title="BMI Distribution", color\_discrete\_sequence=px.colors.sequential.Emrld\_r)

plt.show()

plt = px.histogram(df, x="AlcoholConsumption", title="AlcoholConsumption Distribution",

color\_discrete\_sequence=px.colors.sequential.thermal\_r)

plt.show()

plt = px.pie(df, names="Smoking", title="Smoking Distribution", color\_discrete\_sequence=px.colors.sequential.Viridis)

plt.show()

plt = px.pie(df, names="GeneticRisk", title="GeneticRisk Distribution", color\_discrete\_sequence=px.colors.sequential.Aggrnyl)

plt.show()

plt = px.histogram(df, x="PhysicalActivity", title="PhysicalActivity Distribution", color\_discrete\_sequence=px.colors.sequential.amp\_r)

plt.show()

plt = px.pie(df, names="Diabetes", title="Diabetes Distribution", color\_discrete\_sequence=px.colors.sequential.Agsunset)

plt.show()

plt = px.pie(df, names="Hypertension", title="Hypertension Distribution", color\_discrete\_sequence=px.colors.sequential.Blackbody)

plt.show()

plt = px.histogram(df, x="LiverFunctionTest", title="LiverFunctionTest Distribution", color\_discrete\_sequence=px.colors.sequential.ice)

plt.show()

plt = px.pie(df, names="Diagnosis", title="Diagnosis Distribution", color\_discrete\_sequence=px.colors.sequential.Magenta)

plt.show()

for col in df.columns:

if col is not "Diagnosis":

plt = px.histogram(df, x=col, color = "Diagnosis", title=f"{col} Distribution", color\_discrete\_sequence=px.colors.sequential.Agsunset\_r)

plt.show()

import matplotlib.pyplot as plt

corr = df.corr()

plt = plt.figure(figsize=(10, 10))

sns.heatmap(corr, annot=True, cmap="coolwarm")

X = df.drop("Diagnosis", axis=1)

y = df["Diagnosis"]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

models = {

"CatBoost": CatBoostClassifier(iterations=100, random\_state=42),

"XGBoost": XGBClassifier(n\_estimators=100, random\_state=42),

"LightGBM": LGBMClassifier(n\_estimators=100, random\_state=42)

}

for name, model in models.items():

model.fit(X\_train\_scaled, y\_train)

model\_scores = {}

for name, model in models.items():

y\_pred = model.predict(X\_test\_scaled)

accuracy = accuracy\_score(y\_test, y\_pred)

model\_scores[name] = accuracy

print(f"{name} Accuracy: {accuracy:.2f}")

best\_model\_name = max(model\_scores, key=model\_scores.get)

best\_model\_score = model\_scores[best\_model\_name]

print(f"Best model: {best\_model\_name} with accuracy: {best\_model\_score:.2f}")

Best model: CatBoost with accuracy: 0.92\*

import pickle

best\_model = models[best\_model\_name]

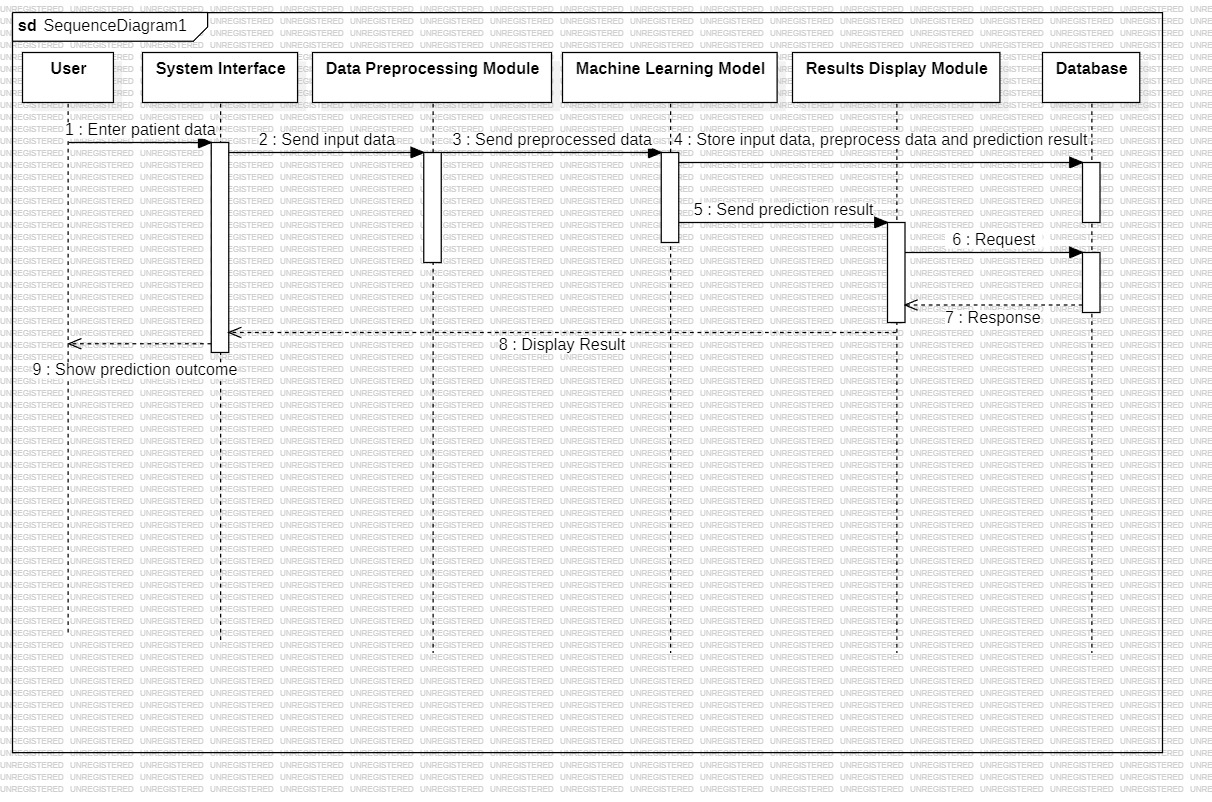
with open('best\_model.pkl', 'wb') as f:

pickle.dump(best\_model, f)

with open('scaler.pkl', 'wb') as f:

pickle.dump(scaler, f)

### Flow charts



The above flowchart gives the brief about Sequential process of Classification and Prediction of Liver disease.

Steps Involved:

* + 1. Collecting of Patient Data
    2. Preprocessing the Data
    3. Train the Model
    4. Make Prediction

## TESTING AND VALIDATION

**6.1.INTRODUCTION**

Testing is a procedure , which uncovers blunders in the program . Programming testing is a basic

component of programming quality affirmation and speaks to a definitive audit of determination,

outline and coding . The expanding perceivability of programming as a framework component and chaperon costs related with a product disappointment are propelling variables for arranged, through testing.

Testing is the way toward executing a program with the plan of finding a mistake. The plan of tests

for programming and other built items can be as trying as the underlying outline of the item itself It

is the significant quality measure utilized amid programming improvement . A mid testing, the program is executed with an arrangement of experiments and the yield of the program for the experiments is assessed to decide whether the program is executing as it is relied upon to perform.

A technique for programming testing coordinates the outline of programming experiments into an all - around arranged arrangement of steps that outcome in fruitful improvement of the product. Keeping in mind the end goal to ensure that the framework does not have blunders, the distinctive

levels of testing techniques that are connected at varying periods of programming improvement are:

Unit Testing is done on singular modules as they are finished and turned out to be executable . It is restricted just to the planner's prerequisites . It centers testing around the capacity or programming

module . It Concentrates on the interior preparing rationale and information structures. It is rearranged when a module is composed with high union.

* Reduces the quantity of experiments
* Allows mistakes to be all the more effectively anticipated and revealed .

## 

## BLACK BOX TESTING :

It is otherwise called Functional testing . A product testing strategy whereby the inward workings of the thing being tried are not known by the analyzer . For instance , in a discovery test on a product outline the analyzer just knows the information sources and what the normal results ought to be and not how the program touches base at those yields . The analyzer does not ever inspect the programming code and does not require any further learning of the program other than its determinations. In this system some experiments are produced as information conditions that completely execute every single practical prerequisite for the program.

This testing has been utilizations to discover mistakes in the accompanying classifications: Incorrect or missing capacities

Interface blunders

Errors in information structure or outside database get to Performance blunders

Initialization and end blunders. In this testing just the yield is checked for rightness.

## WHITE BOX TESTING :

It is otherwise called Glass box , Structural, Clear box and Open box testing .A product testing procedure whereby express learning of the inner workings of the thing being tried are utilized to choose the test information. Not at all like discovery testing, white box testing utilizes particular learning of programming code to inspect yields. The test is precise just if the analyzer comprehends what the program should do. He or she would then be able to check whether the program from its expected objective. White box testing does not represent blunders caused by oversight, and all obvious code should likewise be discernable. For an entire programming examination, both white box and discovery tests .

## INTEGRATION TESTING:

Coordination testing guarantees that product and subsystems cooperate an entirety. It tests the interface of the considerable number of modules to ensure that the modules carry on legitimately when coordinated together. It is characterized as a deliberate procedure for developing the product engineering. In the meantime, reconciliation is happening , lead tests to reveal blunders related with interfaces. Its Objective is to take unit tried modules and assemble a program structure in view of the recommended outline Two Approaches of Integration Testing

1. Non-incremental Integration Testing

2. Incremental Integration Testing

## FUNCTIONAL TESTING:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation , and user manuals. Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted. Invalid Input : identified classes of invalid input must be rejected. Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised. Systems/Procedures: interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. For an entire programming examination , both white box and discovery tests are required. It has been utilizations to produce the experiments in the accompanying cases:

Guarantee that every single free way have been Executed.

Execute every single intelligent choice on their actual and false Sides.

.

In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing . Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

## ACCEPTANCE TESTING :

Acknowledgment testing , a testing method performed to decide if the product framework has met the prerequisite particulars . The principle motivation behind this test is to assess the framework's consistence with the business necessities and check in the event that it is has met the required criteria for conveyance to end clients. It is a pre-conveyance testing in which whole framework is tried at customer's site on genuine information to discover blunders.

The acknowledgment test bodies of evidence are executed against the test information or utilizing an acknowledgment test content and afterward the outcomes are contrasted and the normal ones. The acknowledgment test exercises are completed in stages. Right off the bat, the essential tests are executed, and if the test outcomes are palatable then the execution of more intricate situations are done.

## 6.2.TEST APPROACH

A Test approach is the test system usage of a venture, characterizes how testing would be done. The decision of test methodologies or test technique is a standout amongst the most intense factor in the achievement of the test exertion and the precision of the test designs and gauges. Testing should be possible in two ways

1. Bottom-up approach

2. Top-down approach

### Bottom-up Approach:

Testing can be performed beginning from littlest and most reduced level modules and continuing each one in turn . In this approach testing is directed from sub module to primary module , if the fundamental module is not built up a transitory program called DRIVERS is utilized to recreate the principle module . At the point when base level modules are tried consideration swings to those on the following level that utilization the lower level ones they are tried exclusively and afterward connected with the already inspected bring down level modules.

### Top-down Approach:

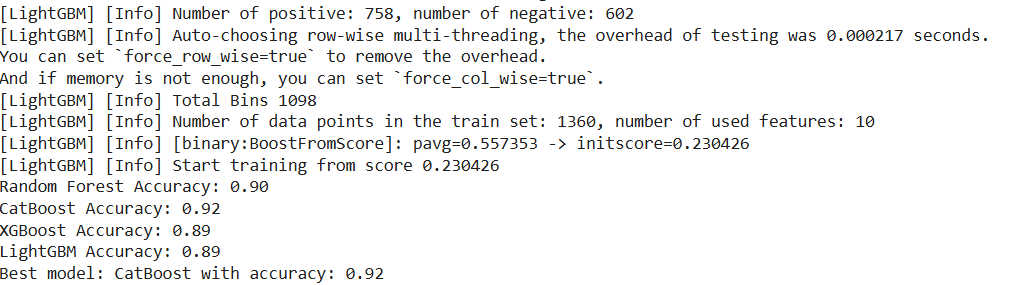
In this approach testing is directed from fundamental module to sub module. in the event that the sub module is not built up an impermanent program called STUB is utilized for mimic the sub module . This sort of testing begins from upper level modules. Since the nitty gritty exercises more often than not performed in the lower level schedules are not given stubs are composed . A stub is a module shell called by upper level module and that when achieved legitimately will restore a message to the calling module demonstrating that appropriate association happened.

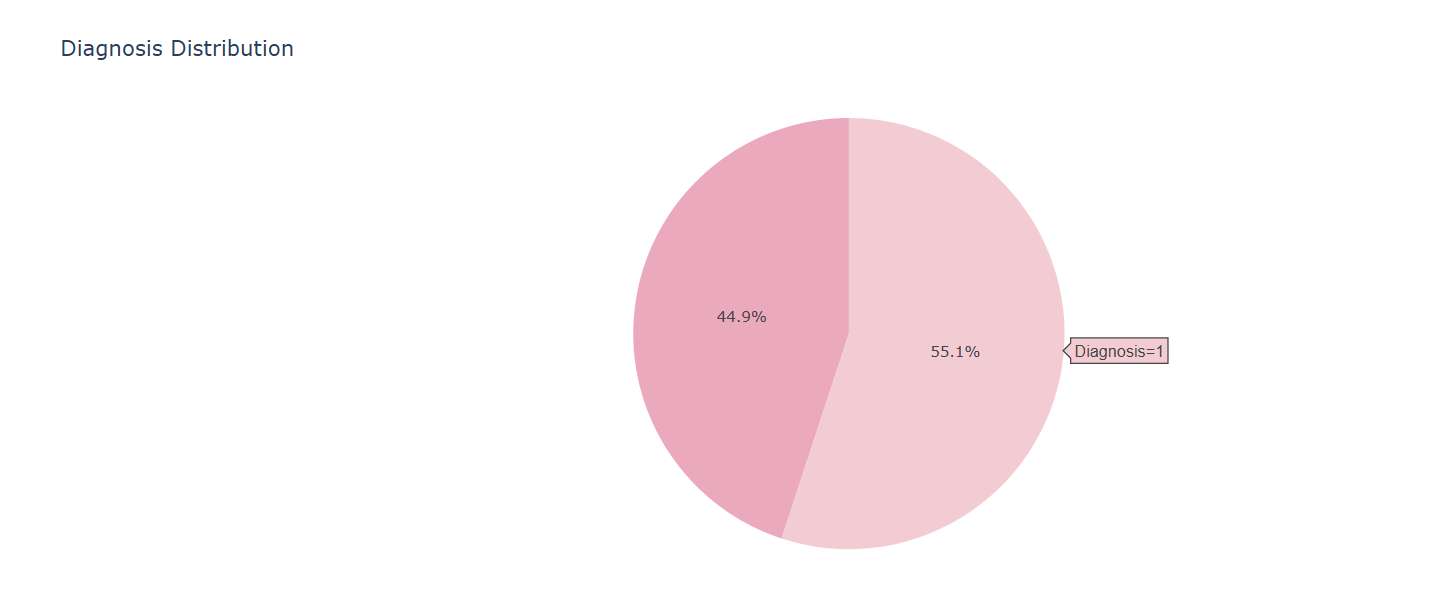
## 6.3.VALIDATION:

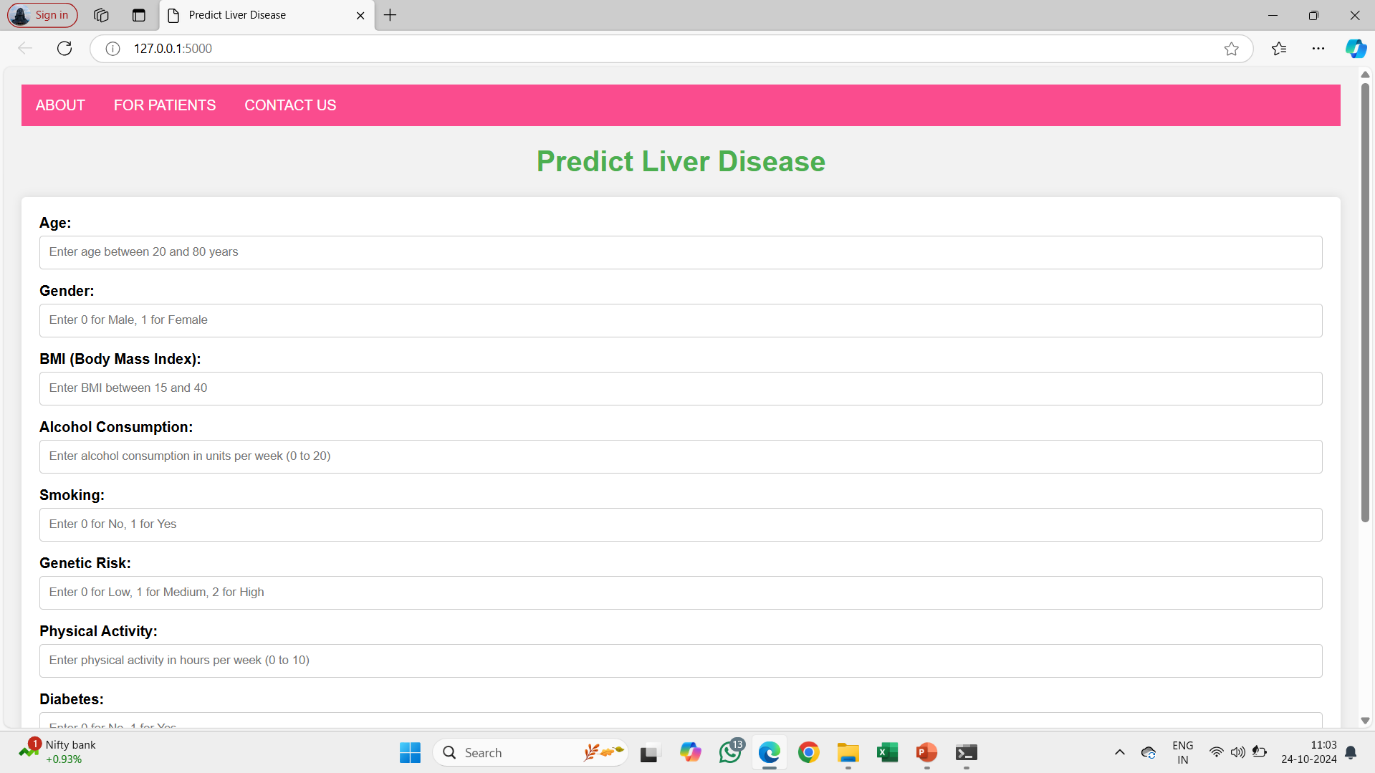
The way toward assessing programming amid the improvement procedure or toward the finish of the advancement procedure to decide if it fulfills determined business prerequisites . Approval Testing guarantees that the item really addresses the customer's issues . It can likewise be characterized as to exhibit that the item satisfies its proposed utilize when sent on proper condition . The framework has been tried and actualized effectively and along these lines guaranteed that every one of the prerequisites as recorded in the product necessities determination are totally satisfied.

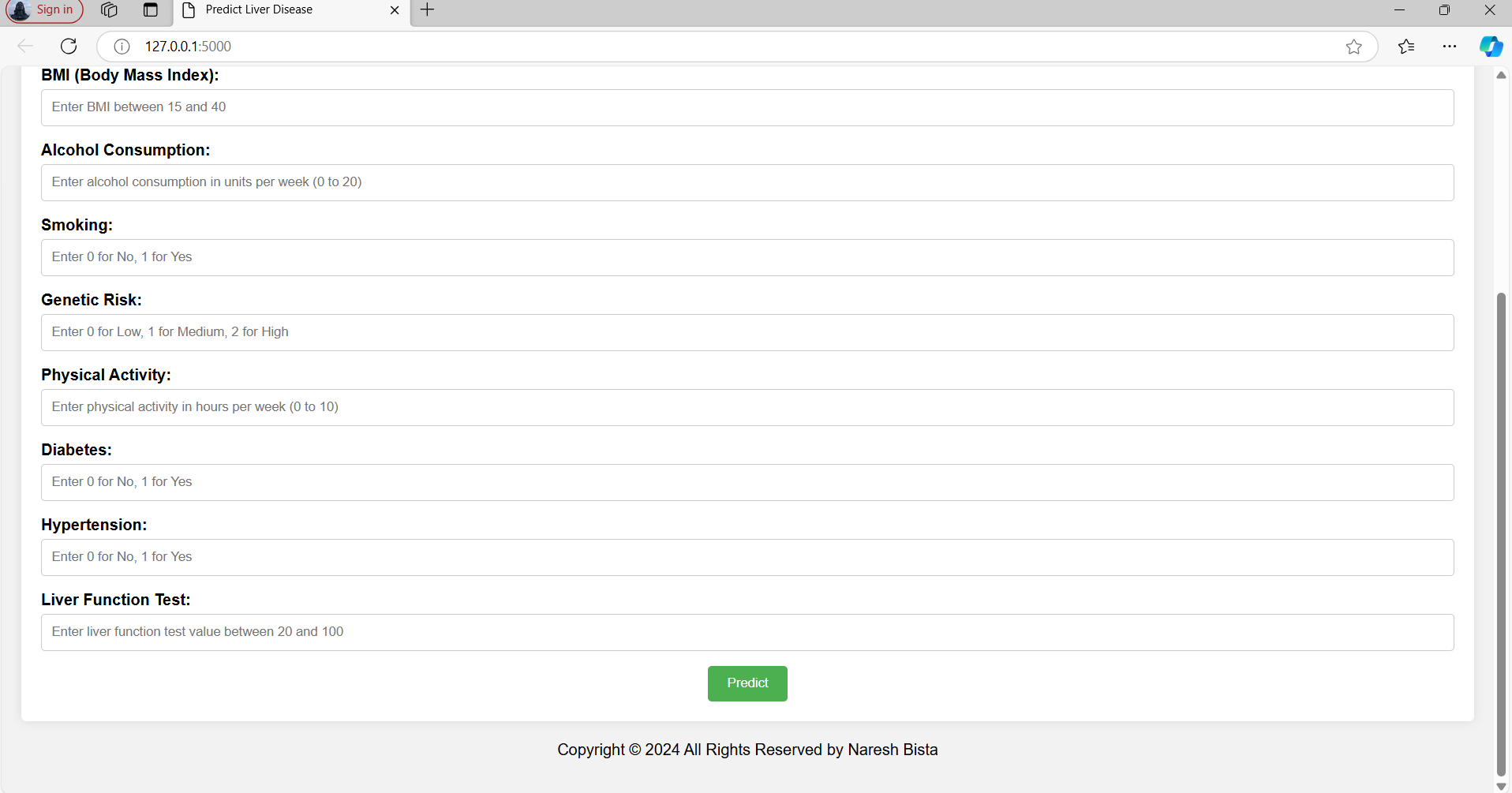
## 7. RESULTS

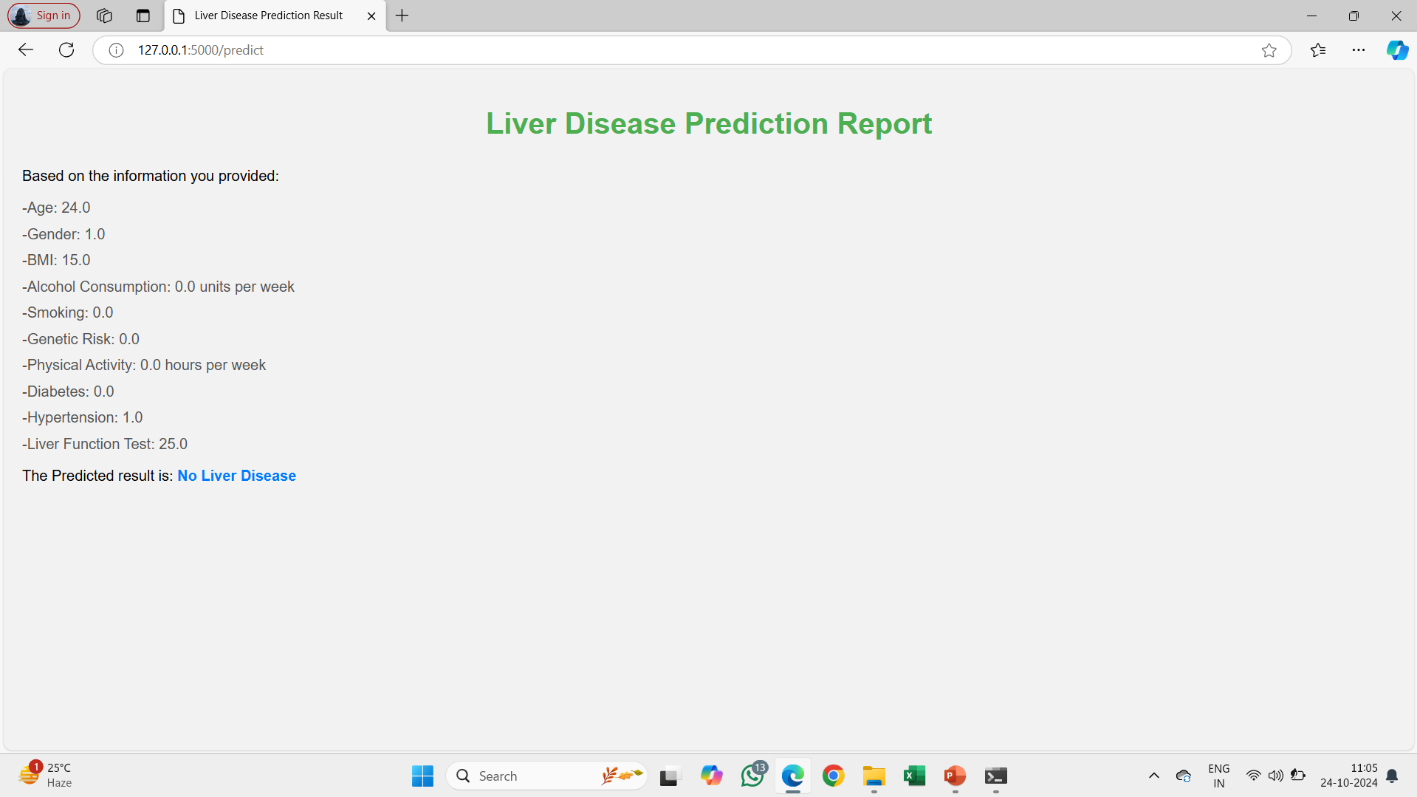
**MODEL ACCURACY**











## CONCLUSION & FUTURE SCOPE

In this project, we developed a robust machine learning model to classify and predict liver disease based on various health-related parameters. By utilizing advanced algorithms such as Decision Tree, CatBoost, XGBoost, and LightGBM, we achieved significant accuracy in predicting the presence of liver disease. Our model effectively processed and analyzed the input data, highlighting the critical factors contributing to liver health, including age, BMI, alcohol consumption, and genetic risk. The interactive web application we built not only enhances user engagement but also facilitates timely diagnosis and intervention, potentially improving patient outcomes.

Through this project, we have demonstrated the efficacy of machine learning techniques in healthcare, showcasing how data-driven approaches can lead to better decision-making and resource allocation in the medical field.

The future scope of this project encompasses several key enhancements that could make it even more impactful in the healthcare field. First, improving the model's accuracy through advanced techniques like hyperparameter tuning, and exploring other machine learning algorithms, could yield higher precision in liver disease predictions. Additionally, incorporating real-time clinical data—such as liver enzyme levels or medical imaging—would make the model more relevant in real-world applications, allowing for dynamic updates and more personalized results. Using larger and more diverse datasets, including data from various regions and demographics, would also enhance the model's generalizability and help identify region-specific risk factors.

## REFERENCES

1.**Sharma, A., & Kumar, P. (2020).** "Prediction of liver disease using machine learning algorithms." *International Journal of Advanced Research in Computer Science and Software Engineering*, 10(5), 75-81.

2. **Singh, R., & Gupta, N. (2020).** "Prediction of liver disease using data mining techniques." *Materials Today: Proceedings*, 22, 2505-2509.

3.**Sah, R., & Patra, M. R. (2019).** "A survey of machine learning algorithms for liver disease prediction and classification." *International Journal of Advanced Computer Science and Applications*, 10(7), 462-468.

4. **Kumar, M., & Patel, S. (2019).** "Liver disease detection using random forest classifier and feature selection." *Procedia Computer Science*, 152, 48-55.

**5. Kumar, S., & Dey, N. (2021).** "A novel approach to liver disease classification using machine learning techniques." *Journal of King Saud University - Computer and Information Sciences*, 33(4), 445-452.  
https://doi.org/10.1016/j.jksuci.2019.11.029

**6. Verma, A., & Gupta, A. (2020).** "Liver disease prediction using ensemble learning models: A comparative analysis." *Soft Computing*, 24, 7093-7102.

7.**Jain, S., & Garg, R. (2019).** "Application of deep learning algorithms for liver disease diagnosis." *Proceedings of the International Conference on Recent Advances in Computing and Software Systems*, 258-265.

8. **Zhang, L., & Li, J. (2018).** "A survey of machine learning algorithms in liver disease prediction." *Journal of Healthcare Engineering*, 2018.