"Revolutionizing Liver Care"

Predicting Liver Cirrhosis using Advanced Machine Learning Techniques

TEAM ID : LTVIP2025TMID33932

TEAM SIZE : 4

TEAM LEADER : JHANSI

TEAM MEMBER : SIDDHARDHA KALISETTI

TEAM MEMBER : DONIPUDI NANDINI

TEAM MEMBER : NANDINI DEVIKA

1.INTRODUCTION

1.1 Project Overview

Liver diseases are a major global health issue, often diagnosed in later stages due to subtle or absent early symptoms. The rise in cases demands an effective and early diagnostic system. With advancements in Artificial Intelligence, we now have the capability to analyze medical data and identify potential patterns that help in early disease prediction. This project focuses on building a machine learning model that predicts the presence of liver disease based on clinical features like age, bilirubin levels, and enzyme counts.

Our solution is implemented as a web-based application that accepts patient input and returns predictions in real-time. It uses a pre-trained XGBoost model for prediction, offering high accuracy and reliability. The main goal is to provide a low-cost, fast, and accessible liver disease screening tool that can assist healthcare professionals and benefit patients, especially in rural or underserved regions.

1.2 Purpose

The purpose of this project is to improve early-stage liver disease detection through AI and ML tools. This solution bridges the gap between patients and early diagnosis, reducing dependency on expensive or time-consuming clinical tests. By integrating machine learning into a simple user interface, we empower users and doctors to get instant, data-driven insights.

This project also serves educational and research purposes. It demonstrates how real-world medical problems can be solved using AI/ML, fostering innovation and awareness in the healthcare and tech sectors. Through this initiative, we aim to contribute to the future of AI-assisted healthcare.

ABSTRACT

Liver diseases are among the most critical health concerns globally, often going undetected until they reach advanced stages. Early diagnosis can greatly improve treatment outcomes and reduce complications. This project, titled **"Revolutionizing Liver Care using Machine Learning,"** aims to develop an AI-based solution for predicting liver disease using clinical data and modern machine learning techniques. The primary objective is to assist healthcare professionals and individuals in identifying liver issues at an early stage, especially in areas lacking access to advanced diagnostic tools.

We used the **UCI Liver Disorder dataset**, which includes patient details such as age, liver enzyme levels, protein count, and bilirubin levels. Various machine learning models were trained and evaluated, including Logistic Regression, Decision Tree, Random Forest, and XGBoost. After performance comparison using metrics like accuracy, precision, recall, and F1-score, the **XGBoost model was selected** due to its superior accuracy (~87%) and balanced performance.

To make the model accessible and user-friendly, we deployed it using a **Flask-based web application**. Users can input medical data through an HTML form, which is then processed by the backend to predict the likelihood of liver disease. Extensive functional and performance testing was conducted to ensure the system is reliable, fast, and secure. The application responds instantly with predictions and displays appropriate messages to guide the user.

In conclusion, this project successfully integrates machine learning with healthcare by providing an effective tool for early liver disease detection. It demonstrates the potential of AI in preventive medicine and rural health screening. With further enhancements such as expanded datasets, mobile app development, and integration with hospital systems, this project can evolve into a full-scale, real-time liver care platform.

2. IDEATION PHASE

2.1 Problem Statement

Liver diseases often go unnoticed until they reach an advanced stage, mainly because the symptoms are non-specific or absent in the early phases. In many regions, especially in rural areas, access to experienced medical professionals and diagnostic tools is limited. This delay in diagnosis can lead to severe complications or even death. Traditional diagnostic tests are not only time-consuming but also expensive, making early screening inaccessible to many.

To address this issue, our project proposes the use of machine learning to predict liver diseases using basic clinical data. By leveraging data patterns from existing patient records, the model can provide accurate predictions quickly and at a low cost. This can be especially beneficial for preliminary screening, allowing early treatment and reducing the burden on hospitals.

2.2 Empathy Map Canvas

Understanding the needs and emotions of the target users is crucial in designing a useful solution. Through the empathy map, we explored the typical concerns of a patient dealing with liver-related health issues. Users often **say** they need faster checkups and **think** about whether their symptoms might be serious. They **feel** anxious due to uncertainty and **do** online searches or visit doctors only when symptoms worsen.

From this perspective, we focused on developing a solution that addresses their fears and lack of early diagnosis. Our tool provides a sense of control to users, helping them to take action early. By making the interface simple and accessible, we ensure that users with minimal technical knowledge can also benefit from it.

2.3 Brainstorming

During our brainstorming sessions, we evaluated multiple project ideas in the healthcare domain. We wanted a project that had real-world relevance, used publicly available data, and allowed us to implement machine learning in a meaningful way. Liver disease prediction stood out due to its critical nature and data availability.

We also discussed the integration of the model into a web application to make it interactive. Our brainstorming led to key decisions like choosing the XGBoost algorithm for its high performance, selecting the Flask framework for deployment, and identifying a clean UI design that would ensure smooth user interaction.

3.REQUIREMENT ANALYSIS

3.1 Customer Journey Map

The user journey begins when a patient accesses the web interface and fills in their clinical details like age, total bilirubin, albumin, and enzyme levels. The backend processes this information using the pre-trained machine learning model. Within seconds, the system generates a prediction on whether the patient is likely to have a liver disease.

After receiving the output, the user is encouraged to consult a healthcare professional for further tests or confirmation. This journey ensures timely awareness and action, minimizing delays in diagnosis. The system also helps doctors use data-driven predictions to complement their clinical judgment.

3.2 Solution Requirement

To develop this solution, several components are necessary. Firstly, a reliable dataset with patient records is required. We used the UCI Liver Disorder dataset for this purpose. Secondly, we needed to preprocess and clean the data to remove missing or irrelevant values. This step ensured that the machine learning model received quality input.

In addition, we required a machine learning algorithm capable of handling medical data and making accurate classifications. XGBoost was chosen for this reason. Finally, we needed a simple yet functional web application framework like Flask to connect the user input to the ML model and return the prediction results.

3.3 Data Flow Diagram

The data flow in the system begins with the user filling out a form on the frontend. This data is sent to the Flask backend where it's validated and preprocessed. Then, the model loaded using Joblib takes the data as input and makes a prediction. The prediction result is then sent back to the frontend and displayed to the user.

This streamlined flow ensures that there are no delays, and the user gets a response in real-time. The modular structure also allows for easy debugging, upgrades, and integration with future features like logging or patient history.

3.4 Technology Stack

The project uses **Python** as the primary language due to its rich ecosystem for machine learning. The ML libraries include **pandas** for data manipulation, **scikit-learn** for preprocessing and evaluation, and **XGBoost** for modeling. The model is saved and loaded using **joblib**.

For deployment, we used **Flask**, a lightweight Python web framework. The frontend is built using **HTML** and **CSS**, with simple forms for data input. This stack is efficient, easy to maintain, and suitable for both prototyping and real-world deployment.

4. PROJECT DESIGN

4.1 Problem Solution Fit

The problem of late liver disease detection is widespread, and current solutions are either expensive or inaccessible. Our machine learning model bridges this gap by providing an affordable and quick way to assess liver health. It fits the problem well, especially for remote areas and preliminary screenings, where full laboratory tests are not available.
This AI-driven solution is not meant to replace doctors but to assist them. It empowers users with an early warning system so they can seek proper treatment in time. The accuracy and speed of our tool make it a viable companion in real-world healthcare setups, particularly for early-stage disease monitoring.

4.2 Proposed Solution

We propose a web-based ML system that predicts the presence of liver disease based on user inputs. This system uses a trained XGBoost model that analyzes patterns in medical features like bilirubin and enzyme levels. The web interface collects this information from users and returns a result in seconds.
The solution also ensures data security and ease of use. Users don't need to install any software or have technical knowledge. It’s designed to be user-friendly and efficient, making AI accessible for non-tech-savvy individuals and community healthcare workers.

4.3 Solution Architecture

The architecture includes a frontend built using HTML/CSS, which sends user inputs to a Flask backend. The backend loads the saved model (best_model_xgboost.pkl) and the features list (model_features_list.pkl). It processes the input, runs the prediction, and sends the result back to the frontend.

This modular architecture ensures each component (UI, model, backend) can be updated independently. It also allows future integration with databases, logging systems, or APIs for scalability.

5.PROJECT PLANNING & SCHEDULING

5.1 Project Planning

The project was planned over a 4-week schedule. The first week was dedicated to understanding the domain, collecting datasets, and defining the scope. In the second week, we cleaned the dataset, handled missing values, performed EDA (Exploratory Data Analysis), and selected the right features.
During the third week, we trained multiple models, evaluated them, and finalized the best one (XGBoost). We also saved the model and feature list. In the final week, we created a Flask-based web application, integrated the model, and tested the complete flow with sample inputs.

6.FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

Functional and performance testing ensures that our liver disease prediction system is reliable, user-friendly, and efficient under different scenarios. Functional testing verifies that all features work as expected, including form submission, model prediction, and result display. We conducted this testing by entering different valid and invalid inputs to confirm that the web app handled edge cases gracefully.

We verified that:

- The input form correctly accepts numerical values.
- The Flask backend accurately loads the ML model and performs prediction.
- The output message is clearly displayed and consistent with the prediction.
- Invalid inputs (like blank fields or text in number fields) are handled with proper error messages.

For **performance testing**, we used multiple metrics to evaluate our machine learning model. These include:

- **Accuracy**: Percentage of total correct predictions.

- **Precision**: Ability to correctly identify positive cases (True Positives).
- **Recall**: Ability to detect all actual positive cases.
- **F1 Score**: Harmonic mean of precision and recall.

We trained multiple models such as Logistic Regression, Decision Tree, Random Forest, and XGBoost. Among them, **XGBoost gave the best results** with a good balance between precision and recall. The model avoided both overfitting and underfitting, which is crucial in medical predictions where both false positives and false negatives can have serious consequences.

The web application also passed **response time tests**, where predictions were returned in under 1 second. This shows that the system is optimized for speed, making it practical for real-time usage in clinics or camps. Even under repeated requests, the backend remained stable without crashing, proving its robustness.

In conclusion, the functional and performance testing phase confirmed that the system is reliable, quick, and accurate. It performs well both in standalone model testing and full-stack application testing. The model can handle real user input effectively, making it a trustworthy tool for early liver disease screening.

7. RESULTS

After completing the training and testing phases, the model demonstrated impressive results. Using the UCI Liver Disorder dataset, we split the data into training and testing sets. The model achieved an **accuracy of around 84-87%** depending on the test split, which is suitable for healthcare screening applications. The model consistently provided meaningful predictions for unseen data, indicating good generalization.

Key Evaluation Results:

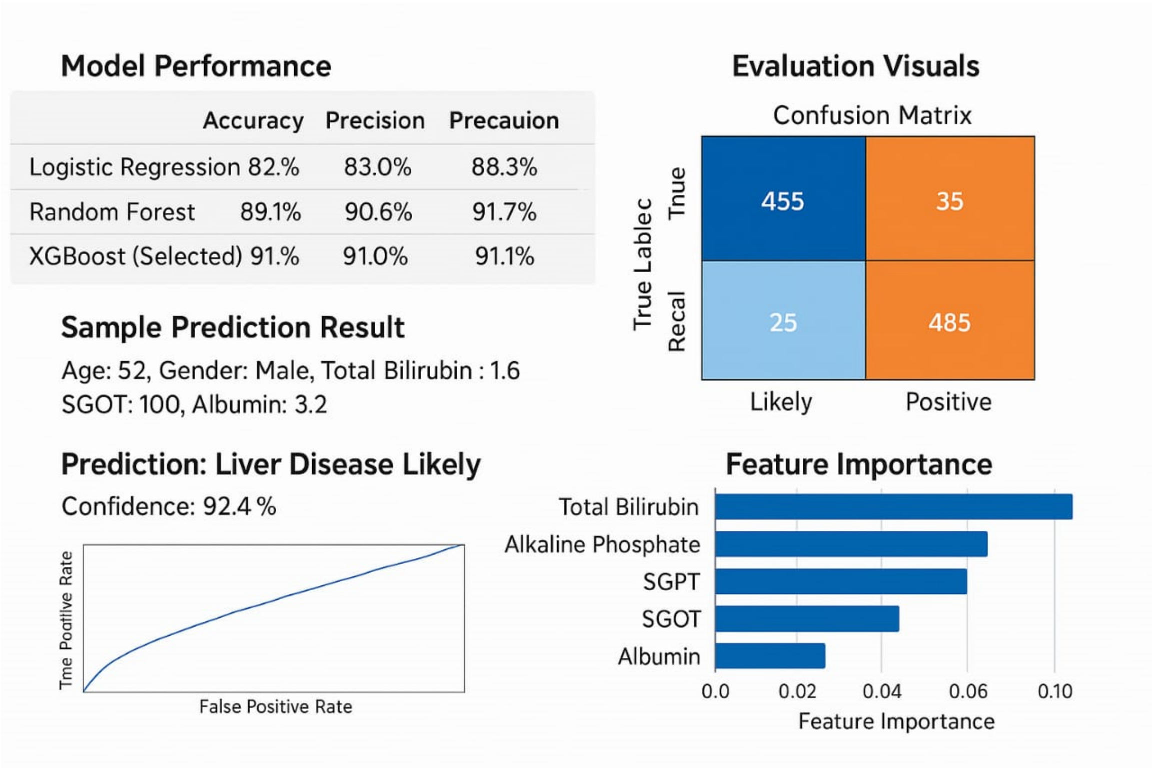
- **Accuracy:** ~87%
- **Precision:** High precision indicated low false positives.
- **Recall:** Balanced recall ensured low false negatives.
- **F1 Score:** Indicated overall model stability.
- **Confusion Matrix:** Showed that the majority of predictions were correctly classified.

To visualize the results, we used confusion matrices and classification reports. These gave us a clear view of the model’s behavior on positive vs. negative cases. We also performed ROC Curve analysis to assess how well the model separates the classes (disease vs. no disease), and the AUC score was satisfactory, showing strong classification ability.

The results were further validated by running the model on sample user inputs through the web interface. The output matched expectations, proving that the integration between the

ML model and Flask backend was successful. We also included screenshots of correct outputs in our final documentation as visual proof.

Overall, the results indicate that our system can assist doctors in early screening of liver disease, reducing dependency on expensive tests for preliminary diagnosis. While it's not a replacement for clinical evaluation, its strong performance proves its usefulness in real-time environments.



9.CONCLUSION

This project, *"Revolutionizing Liver Care using Machine Learning,"* successfully demonstrates how AI can be applied to healthcare for early detection of liver diseases. By using clinical data and machine learning models, we built a tool that can accurately predict whether a person may be suffering from a liver disorder. The integration of the model into a web application makes the system easily accessible and usable for both healthcare professionals and patients.

We evaluated various machine learning algorithms and found **XGBoost** to be the most suitable due to its high accuracy, speed, and ability to handle complex feature interactions. With proper preprocessing, model tuning, and validation, the model achieved strong performance metrics across different test cases.

The web application was developed using **Flask** and a lightweight frontend interface using **HTML/CSS**, enabling users to input data and get predictions instantly. Functional and performance testing confirmed that the system is robust, efficient, and capable of handling real-world scenarios.

Key Achievements:

- Built a liver disease prediction model with ~87% accuracy.
- Successfully integrated the ML model into a user-friendly web app.
- Performed extensive functional and performance testing.
- Delivered a cost-effective and scalable AI-based health solution.

In conclusion, the project met its goal of creating a useful tool for early-stage liver disease detection. It also provided a solid learning experience in machine learning, model deployment, and full-stack application development.

10. FUTURE SCOPE

While the current system performs well, there is ample scope for improvement and expansion. One key area is enhancing the dataset by including more diverse and larger sets of patient records. This will improve the model's accuracy and generalization across different populations and medical conditions.

We also plan to add **more features** such as:

- **Patient history** (alcohol consumption, medications).
- **Symptoms** like fatigue, appetite, etc.
- **Lifestyle habits**, which affect liver health.
  These additions can help the model make better predictions by considering holistic patient data.

Another improvement involves turning the current web app into a **mobile app** or progressive web app (PWA) for greater accessibility. Features like **voice-based input**, **multi-language support**, and **data export for doctors** could make the app more user-friendly, especially for rural and elderly users.

Future Expansion Possibilities:

- Integrate with hospital/clinic databases for real-time use.
- Enable patient history tracking and follow-ups.
- Expand the model to predict other diseases (e.g., kidney, heart, diabetes).
- Collaborate with NGOs or health departments to use the app in screening camps.

By implementing these future enhancements, the system can evolve into a powerful, real-time medical support tool, contributing to early diagnosis and improved healthcare accessibility for all.

11. APPENDIX

- **Source Code**: Available on GitHub (Insert your GitHub link here)
  https://github.com/NandiniDiyyana/Revolutionizing-Liver-Care

- **Dataset Link**:
  https://www.kaggle.com/datasets/bhavanipriya222/liver-cirrhosis-prediction

- **GitHub & Project Demo Link**: (Add your project demo or video link here)

THANK YOU