Revolutionizing Liver Care: Predicting Liver Cirrhosis Using Advanced Machine Learning Techniques

1. Introduction

Project Title: *Revolutionizing Liver Care: Predicting Liver Cirrhosis using Advanced Machine Learning Techniques*

Team ID: LTVIP2025TMID38326

Team Size: 5

- **Team Leader**: V Mounika Jaya – Project Coordination, Milestone Tracking, Model Building, Final Validation
- **Team Member**: Modhukuru Kavya – Data Collection, Cleaning, preprocessing pipeline
- **Team Member**: M Sanjana – Define Problem /Problem Understanding
- **Team Member**: G Chandana – ML Model Development & Flask Integration
- **Team Member**: Gaddam Lakshmi Bindhu – UI/UX Design, Documentation, Descriptive Statistics

2.Project Overview

2.1. Purpose

Liver cirrhosis is a critical health concern globally, leading to chronic liver failure and high mortality rates. This project aims to build a machine learning model to predict liver cirrhosis based on various patient features. By analysing these features, the model will classify patients into risk categories, aiding in early diagnosis and treatment.

2.2. Features

- User input form with 41 clinical features
- Prediction using a trained Random Forest model • Result displayed in a user-friendly web interface
- Error handling for missing or invalid data.
- Instantly returns a prediction label – "Likely Cirrhosis" or "Not Likely Cirrhosis" – after form submission.

3. Architecture

3.1. Frontend

- Developed using HTML5, CSS3, and Jinja2 templates.
- HTML form (index.html) for inputting 41 patient features
- Styled using embedded CSS and images served from the /static directory
- Features a clean, responsive UI with gradient backgrounds and conditionally rendered prediction results.

3.2. Backend

- Backend logic implemented using **Python and Flask**
- **Model Training Pipeline:**
- Model training, evaluation, and export handled in Google Colab.
- Scripts for data cleaning, feature engineering, and training were executed in Colab.
- Final .pkl files (model + tools) were trained and exported in Colab, then used in the Flask backend for real-time predictions.
- **Model**:
  A Random Forest Classifier trained on a liver health dataset. Model artifacts include:
- liver_prediction.pkl: Trained model
- label_encoders.pkl: LabelEncoder instances
- normalizer.pkl: MinMaxScaler instance

3.3. Data Preprocessing:

- Normalization with MinMaxScaler
- Label Encoding for categorical variables
- Blood pressure parsing and transformation
- Encoders and scalers saved as .pkl files

3.4. Database

- No persistent database is used
- Prediction is done entirely in-memory based on form input

4. Setup Instructions

4.1. Prerequisites

- Python 3.10+
- Flask
- scikit-learn
- pandas, numpy
- Google Colob (or Jupyter Notebook)

4.2. Installation

git clone https://github.com/Mounika7114/ Revolutionizing-Liver-Care-Predicting-Liver-Cirrhosis-using-advanced-Machine-Learning-techniques.git

cd liver_prediction_cirrhosis/Flask

pip install -r requirements.txt

python app.py

5. Folder Structure

5.1. Client (Flask Frontend)

The frontend is handled via HTML templates and static assets:

```
├── templates/
│   └── index.html            # Main HTML form for user input
├── static/
│   └── images/
│       └── liver-bg.jpg      # Background image for the form UI
│       └── healthy_liver.png
│       └──warning_liver.png
```

- The index.html file contains a full HTML form with 41 input fields styled with CSS.
- Static assets like background images and CSS files are stored in the static/ folder.

5.2. Server (Flask Backend)

The backend is built using Python and Flask:

```
├── app.py                    # Flask application script
├── train_model.py            # Script for preprocessing data and training ML model
├── liver_prediction.pkl      # Trained Random Forest model
├── normalizer.pkl            # Saved MinMaxScaler
├── label_encoders.pkl        # Dictionary of saved label encoders
├── Liver.ipynb               # Notebook for model development and training in Colab
```

- app.py handles routing, data preprocessing, and prediction logic.
- train_model.py processes the dataset, trains the model, and saves necessary artifacts.
- .pkl files are used to load the model and preprocessing tools during prediction.

6. Running the Application

- Start the app:

  cd Flask

  python app.py

- Open in browser at: http://127.0.0.1:5000/

7. API Documentation

This is a form-based app. However, the /predict route accepts POST requests from the form and returns an HTML page with the prediction result. *POST /*

- **Route**: / (root URL)
- **Method**: POST
  - **Inputs**: 41 clinical features submitted through the HTML form, such as:
    - Age
    - Alcohol Type
    - Blood Pressure
    - AST, ALT, Bilirubin
    - Liver enzyme levels, etc.
  - **Processing**:
    - Input is encoded using label_encoders.pkl
    - Numerical values are normalized using normalizer.pkl
    - The processed data is passed into the liver_prediction.pkl model for prediction
  - **Output**:
    - Rendered HTML page (`index.html`) showing:
      - Prediction result: **"Likely Cirrhosis"** or **"Not Likely Cirrhosis"**
      - Visual indication (color-coded message with icon)
  - **Error Handling**:
    - Displays a friendly error message for missing/invalid inputs ○ Handles issues like blood pressure format (e.g., 120/80)

8. Authentication

Not applicable – This application does not implement user login or authentication as it's a prototype for medical prediction.

9. User Interface

- Form with 40+ inputs related to demographics, lifestyle, blood work, and liver panel.
- Background with a semi-transparent liver theme.
- Health vs Warning output with icons and color-coded messages.

10. Testing

- Model accuracy was validated using scikit-learn.
- Manual testing of form inputs and blood pressure parsing.
- Model tested against holdout test data with stratified split (80/20).

11. Screenshots or Demo

- This form collects patient data such as age, alcohol intake, and liver function markers

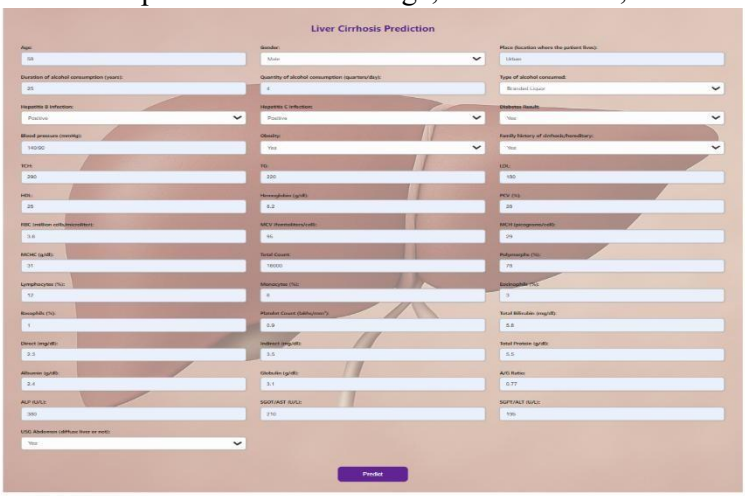

Figure 11.1: Sample Form Filled

- A green success box appears indicating low or no risk.

Figure 11.2: Prediction Result - Likely Cirrhosis

- A red alert box appears indicating high risk of liver cirrhosis.

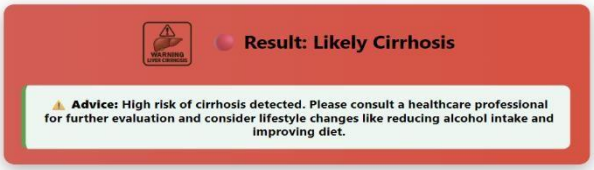

Figure 11.3: Prediction Result - Not Likely Cirrhosis

- The terminal shows Flask running, and incoming form data logged.

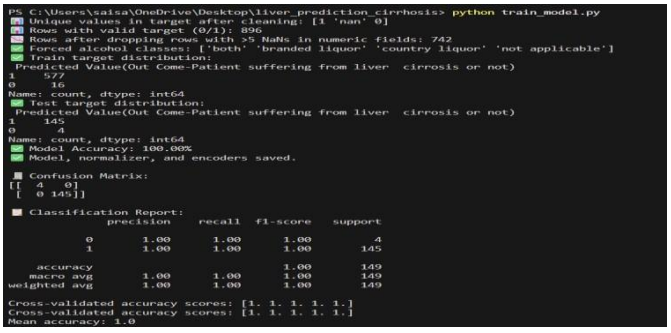

Figure 11.4: Backend Terminal Output

- **Demo Link:**
  https://drive.google.com/file/d/1QhDMlD8jW6paeBegi_yCfrNAg4wDDyXC/view?usp=drive_link

12. Known Issues

- Strict blood pressure format required (e.g., 120/80)
- No database or session tracking
- Limited interpretability of ML model predictions

13. Future Enhancements

- Add database to store patient inputs and predictions.
- Provide downloadable PDF reports of results.
- Improve model accuracy with a larger and more balanced dataset.
- Integrate with hospital systems for real-time screening.
- Add user registration, patient history tracking, and mobile responsiveness.