Artificial Intelligence And Machine Learning Project Documentation

1. Introduction

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

Team Members:

- Team Leader: Padamati Ajay Babu

- Team Member: Sania Firdous Shaik

- Team Member: Tarun Kumar Jetti

2. Project Overview

Purpose: This project aims to develop a machine learning-based predictive model using Flask for the early detection of liver cirrhosis from patient clinical data. The model helps identify at-risk individuals, enabling early medical intervention and better patient care.

Features:

- Upload patient data via a Flask web interface

- Predict cirrhotic vs. non-cirrhotic status

- Show model accuracy and prediction results

- Lightweight, responsive design for real-time use

- Flask-based API structure for future integration

3. Architecture

Frontend: Developed using HTML/CSS with Flask's Jinja2 templates to create forms where users can input patient data and receive predictions dynamically.

Backend: Implemented in Python with Flask. The backend loads the trained machine learning model and handles prediction logic. It includes routes for form handling and result display.

Database: No database used in the current version. All predictions are made from real-time form input. In future, MongoDB or MySQL could be added for storing patient records.

4. Setup Instructions

Prerequisites:

- Python 3.x

- Flask

- Required Python libraries: pandas, scikit-learn, xgboost, numpy, pickle

Installation: Clone the repository, install dependencies, and run the Flask application to start the project locally.

5. Folder Structure

Client: Flask Templates & Static Files -

templates: HTML templates using Jinja2

static: CSS, JS, and images

Server: - app.py: Flask application and routing

- model: Contains the saved ML model (e.g., model.pkl,scaler.pkl)

- data: Sample datasets or test inputs

6. Running the Application

Provide commands to start the Flask application locally and access it in the browser at https://revolutionizing-liver-care-predicting-9pdx.onrender.com

7. API Documentation

Document all endpoints exposed by the backend.

Endpoint: /predict

Method: POST (form submission)

Parameters: Form fields with patient clinical data

Response: Prediction result with confidence score

8. Authentication

Explain how authentication and authorization are handled.

Authentication is not included in this version. Future enhancements may include login for doctors or secure access using Flask-Login or JWT.

9. User Interface

Provide screenshots or GIFs showcasing different UI features.

The UI includes:

- A form to enter patient data

- A results page showing predictions

- Basic layout using HTML/CSS with Flask

10. Testing

Describe the testing strategy and tools used.

Testing includes:

- Manual testing with real and sample patient data

- Validation using accuracy, precision, recall, and confusion matrix

11. Demo

link to a demo to showcase the application.
https://drive.google.com/file/d/1hd5pKQpCF_3lltz6Nfl546EgqO_LUQy-/view?usp=drivesdk

12. Known Issues

- No authentication or user data storage

- No database integration yet

- Basic UI; can be improved for production use

13. Future Enhancements

- Add authentication and user dashboards - Integrate MongoDB for storing patient records - Convert to REST API for wider integration - Enhance UI using frontend frameworks like React - Deploy on cloud platforms (Heroku, Render, etc.)