PROJECT

Project Title :

Revolutionizing Liver Care: An Intelligent ML-Driven System for Early Detection and Prognosis of Liver Cirrhosis.

Team Name :

CirrhoSaviors

Team Members :

- Bhumika Sunayana Simhadri
- Bandi Yaswanth Reddy
- Bandi Satwika
- Bonam Srilakshmi

Phase-1 : Brainstorming & Ideation

Objective:

To develop an AI-powered system for early prediction of liver cirrhosis using machine learning algorithms that assists healthcare professionals by enhancing diagnostic accuracy, reducing time to diagnosis, and enabling preventive liver care.

Key Points:

1. Problem Statement:

- Liver cirrhosis, a chronic liver disease, often progresses silently and is diagnosed at advanced stages, limiting treatment effectiveness.
- Traditional diagnostic approaches like liver biopsy are invasive, time-consuming, and require expert interpretation.
- This project aims to automate cirrhosis prediction based on clinical features using machine learning models, enabling earlier and more accessible diagnosis.

2. Proposed Solution:

- A machine learning-based system that analyzes patient data (e.g., bilirubin, albumin, INR, enzyme levels) and predicts the likelihood of cirrhosis.
- Integrates classification models such as Random Forest, Support Vector Machine, Logistic Regression, and Decision Tree for accurate prediction.

3. Target Users:

- Healthcare providers (doctors, liver specialists) for diagnostic support.
- Hospitals and screening centers for mass liver health analysis.
- Telemedicine platforms for remote monitoring and prediction.
- Health researchers analyzing liver disease progression.

4. Expected Outcome:

- A deployable ML application capable of predicting liver cirrhosis risk using clinical data.
- Enhanced decision-making for early treatment and management.
- Reduction in dependency on invasive diagnostic procedures.

Phase-2 : Requirement Analysis

Objective:

Define the technical and functional requirements for the HematoVision application.

Key Points:

1. **Technical Requirements:**

- **Programming Language: Python**
- **Python Packages:** NumPy, Pandas, Scikit-learn, Matplotlib, SciPy, Seaborn, TensorFlow, Flask
- **Frameworks:** Flask for web integration, TensorFlow for deep learning
- **Pre-trained Model:** VGG16 (used for transfer learning)
- **Development Tools:** Command Line (pip install)

2. **Functional Requirements:**

- Ability to **upload microscopic blood cell images** through the web interface.
- Classify blood cells into types like **eosinophils, lymphocytes, monocytes, and neutrophils** using a trained model.
- Display **classification results** along with prediction confidence.
- Provide an easy-to-use interface for doctors, students, and lab technicians.

3. **Constraints & Challenges:**

- Handling imbalanced data across different blood cell classes.

- Managing low-quality or blurry microscope images that affect prediction accuracy.
- Optimizing model size and performance for faster processing and easy deployment.
- Ensuring the web interface is responsive and user-friendly on all devices.

Phase-3: Project Design

Objective:

To define the system architecture and design user interaction flow.

Key Points:

1. **User Interface (UI):**
    - Built using HTML, CSS
    - Allows users to enter clinical data (e.g., bilirubin, albumin)
    - Sends data to the Flask server

2. **Flask Application (Backend):**
    - Receives input from the UI
    - Applies normalization (normalizer.pkl)
    - Loads trained model (best_model.pkl)
    - Returns prediction result to the UI

3. **Machine Learning Model Layer:**
    - Trained using multiple algorithms (Random Forest, XGBoost, etc.)
    - Random Forest chosen based on performance
    - Saved as .pkl file for use in Flask app

4. **Dataset Layer:**
    - CSV file containing liver patient data
    - Includes preprocessing: cleaning, encoding, feature selection
    - Used for training and testing models

Phase 4 : Project Planning

Objective:

Plan and distribute development tasks for timely and efficient project completion.

Sprint	Task	Priority	Duration	Deadline	Assigned to	Dependencies	Expected outcome
Sprint 1	Environment Setup & Data Preprocessing	🔴 High	3 hours	Day 1	Member 1	Python,Pandas	Cleaned data set
Sprint 1	Exploratory Data Analysis	🔴 High	2 hours	Day 1	Member 2	Dataset ready	Insightful plots
Sprint 2	Model Training (LR, SVM, RF, DT)	🔴 High	5 hours	Day 2	Member 3	Preprocessed data	Trained models
Sprint 2	Flask Web Integration	🟡 Medium	3 hours	Day 2	Member 4	Trained Model	Working web app
Sprint 3	UI Testing & Optimization	🟡 Medium	2 hours	Day 2	Member 2 & 3	Web App ready	Responsive UI
Sprint 3	Deployment & Presentation	🟢 Low	1 hour	End of Day 2	Entire Team	Complete System	Ready for Demo

Phase-5: Project Development

Objective:

Implement liver cirrhosis prediction models and web interface.

Key Points:

1. Technology Stack:

- **Frontend:** HTML/CSS via Flask templates
- **Backend:** Flask (Python)
- **ML Models:** Logistic Regression, SVM, Decision Tree, Random Forest
- **Data:** UCI Liver Cirrhosis Dataset
- **Visualization:** Seaborn, Matplotlib

2. Development Process:

- Performed data cleaning and feature engineering.
- Trained and validated multiple classification models.
- Chose best-performing model based on accuracy and F1-score.
- Integrated model into Flask web app.
- Designed a clean interface for prediction and risk display.

3. Challenges & Fixes:

- **Class imbalance** — Solved using stratified sampling and SMOTE.
- **Overfitting** — Handled using pruning in trees and regularization.
- **Data quality issues** — Resolved by imputing missing values.
- **UI glitches** — Fixed by responsive design and CSS tuning.

Phase-6: Functional & Performance

Objective:

Ensure robust and accurate functioning of the system through testing.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Normal liver input	Predicts "No Cirrhosis"	✅ Passed	Tester 1
TC-002	Functional Testing	Cirrhosis data input	Predicts "Cirrhosis"	✅ Passed	Tester 2
TC-003	Performance Testing	Bulk input test	All processed in < 3s	⚠ Slightly Delayed	Tester 3
TC-004	UI Responsiveness	Mobile layout test	Fully responsive	✅ Passed	Tester 2
TC-005	Deployment Testing	Live demo test	Model predicts online	🚀 Deployed	DevOps