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.

Team Members:

- Durga Bhavani Mirla
- Nalabotu Pardha Naga Venkata Sai
- Navya
- Chakadhar

2. Project Overview:

The project aims to predict liver cirrhosis using clinical data, enabling early diagnosis and timely medical intervention. A machine learning model is integrated with a Flask web interface for user-friendly predictions.

2.1 Purpose:

- Interactive UI with real-time prediction
- Pre-trained ML model integration
- Default values pre-filled based on healthy patient record
- Emojis for easy prediction output display

2.2 Features:

- Error handling for missing or invalid data.
- Instantly returns a prediction label - "Cirrhosis" or "No Cirrhosis"after form submission.
- User input form with 41 clinical features
- Prediction using a trained KNN model .
- Result displayed in a user-friendly web interface

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 KNN trained on a liver health dataset**.**

Model artifacts include:

liver_prediction.pkl: Trained model

3.3. Data Preprocessing:

- Normalization with MinMaxScaler
- Label Encoding for categorical variables
- 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 Requirements:

4.1.Prerequisites:

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

4.2.Installation:

https://github.com/Pnvsai888/Revolutionizing-Liver-Care-Predicting-Liver-Cirrhosis-using-Advanced-Machine-Learning-Techniques/tree/main

5. Folder Structure:

```
Liver_Cirrhosis_Predictor/
│
├── app.py                    # Flask backend
├── templates/
│   └── index.html            # Web form UI
├── static/
│   ├── style.css             # Custom styles
│   └── liver-background.jpg  # Background image
├── models/
│   ├── best_model.joblib
│   ├── normalizer.joblib
│   └── model_features.joblib
├── Final_Liver_Cirrhosis_Cleaned.csv
└── requirements.txt
```

6. API Documentation:

Accepts form data and returns a prediction result (No Cirrhosis or Cirrhosis Detected).

7. User Interface:

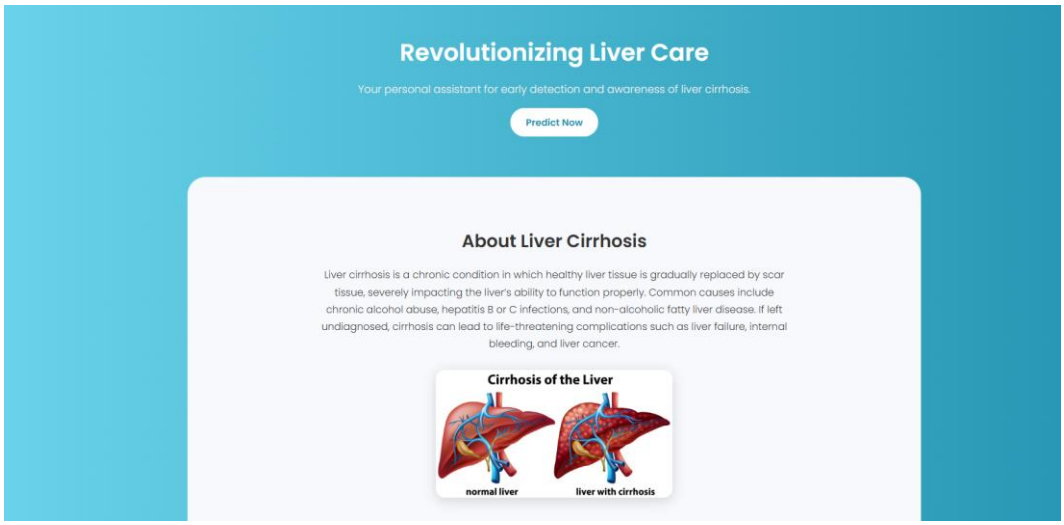
- Responsive layout and clean inputs
- Results shown using intuitive emojis
- Modern design with liver-themed background.

8. Testing:

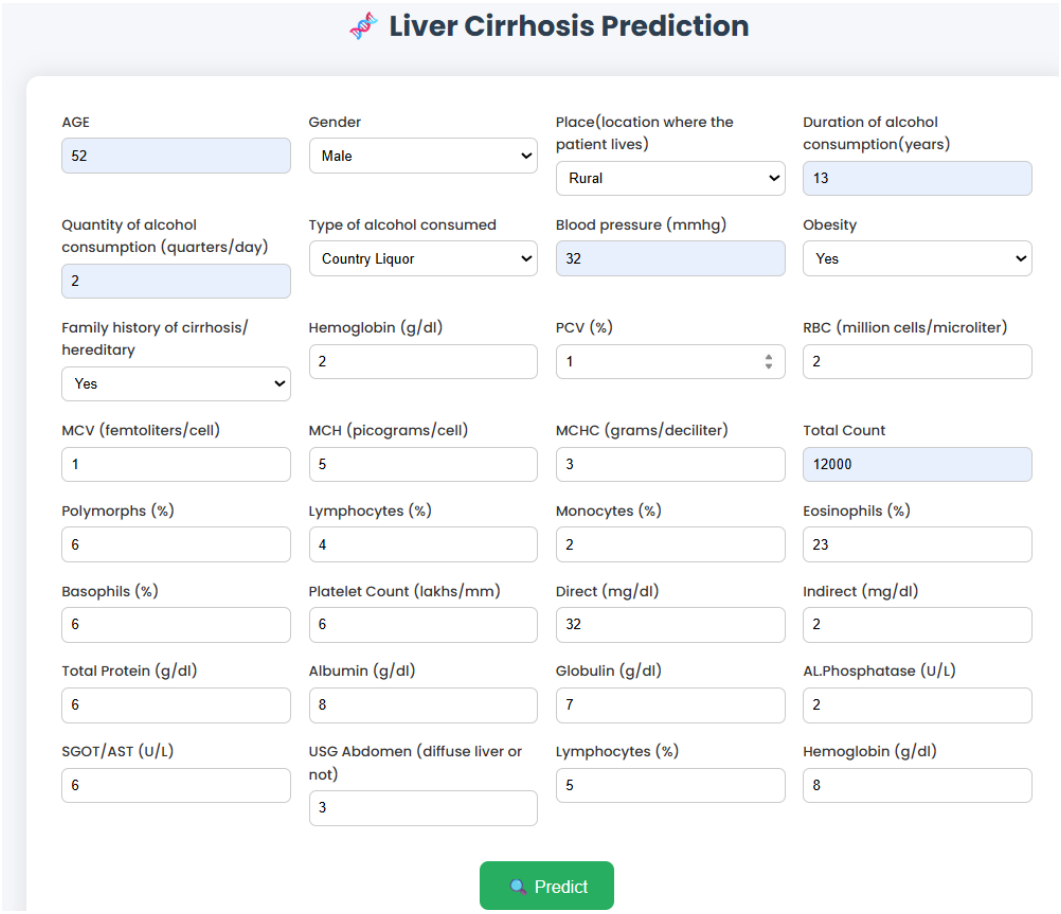
- Manual testing via web interface
- Model evaluated using accuracy and confusion matrix
- Verified prediction correctness using known data samples

9. Screenshots or Demo

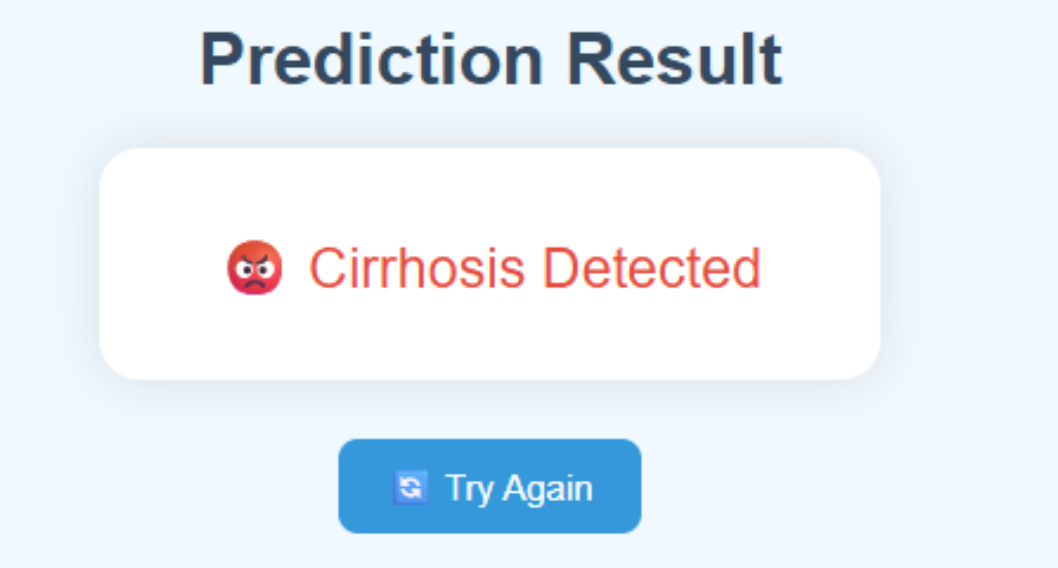
Below are screenshots and visual outputs demonstrating the application's functionality and analysis results



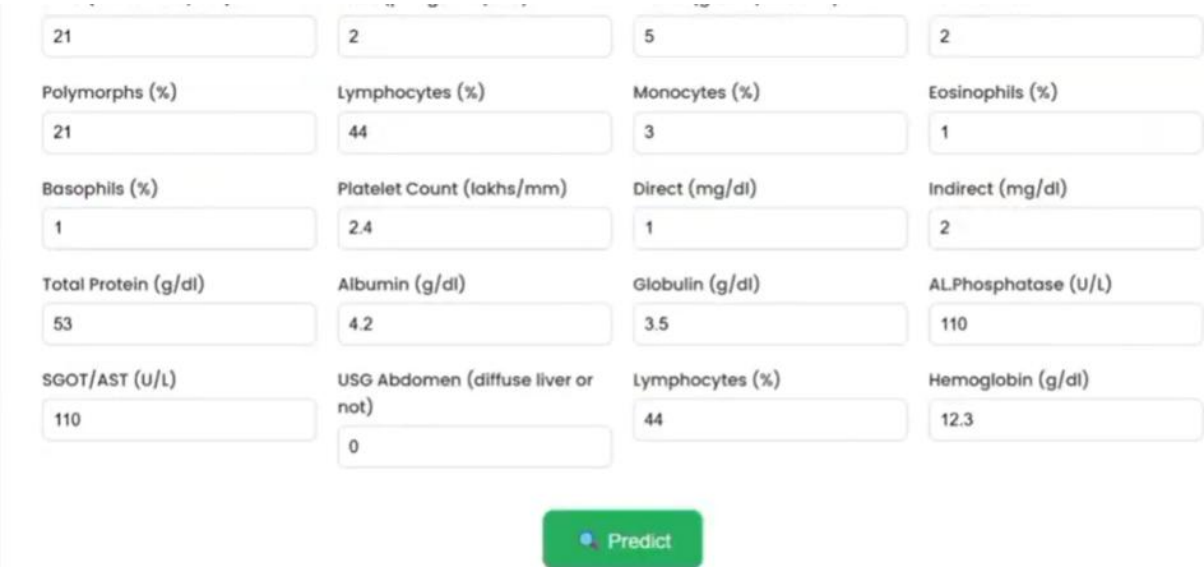
9.1 Home Page - About Liver Cirrhosis Section



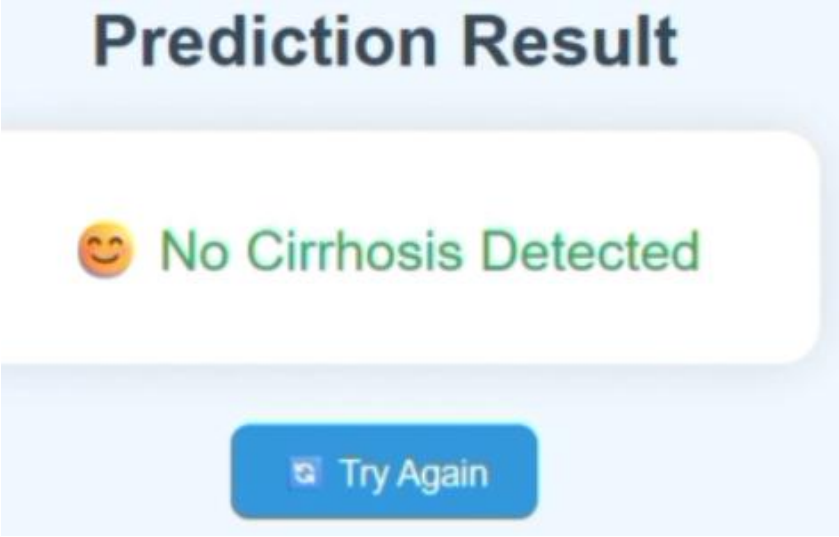
9.2 Prediction Form with Default Values



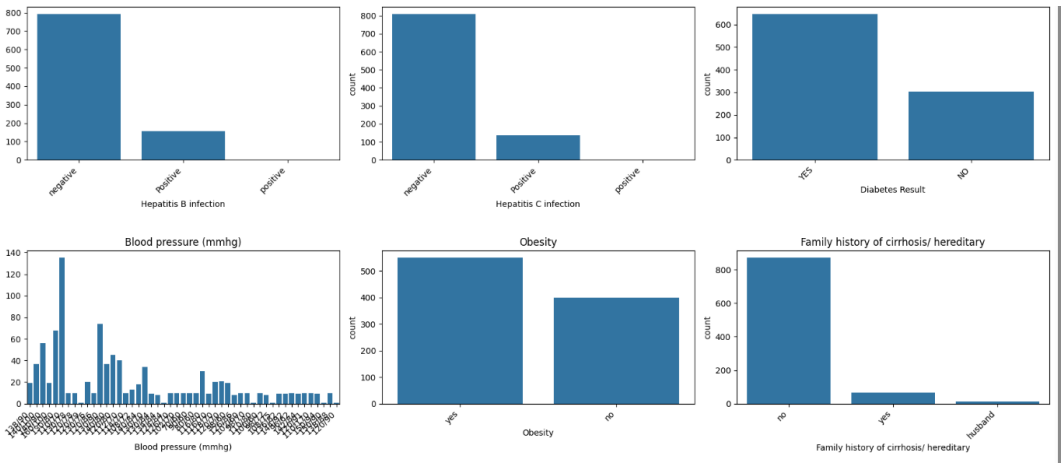
9.3 Prediction Result - Cirrhosis Detected



9.4 Other input Values



9.5 No cirrhosis Detected



9.6. EDA - Categorical Feature Distributions

```
Train score with tuned model: 0.9169960474308301
Test score with tuned model: 0.8631578947368421
Optimal hyperparameters for KNN: {'n_neighbors': np.int64(5)}
Accuracy on test set: 0.86
Confusion Matrix (KNN):
 [[ 49  19]
 [  7 115]]
Classification Report (KNN):
              precision    recall  f1-score   support

           0       0.88      0.72      0.79        68
           1       0.86      0.94      0.90       122

    accuracy                           0.86       190
   macro avg       0.87      0.83      0.84       190
weighted avg       0.86      0.86      0.86       190
```

9.7. Model Evaluation Report and Confusion Matrix

10. Known Issues

- No authentication system implemented
- Lacks database storage
- Only form inputs accepted (no file upload or image data)

11. Future Enhancements

- Add login and session management
- Host on Render/Heroku with database integration
- Enable file uploads for batch prediction
- Use React or Vue for frontend enhancement