

Project Documentation

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 Member: GURRAM SHIVA PRASAD

2. Project Overview

Purpose:

This project aims to develop a predictive web application that uses machine learning to assess whether a patient is at risk of liver cirrhosis based on medical inputs such as blood parameters and alcohol consumption history. The tool is intended to aid in early detection and decision support for healthcare providers.

Features:

- User-friendly input form for patient health parameters
- Accurate liver cirrhosis prediction using trained ML models
- Result page displaying prediction outcome
- Model trained using Random Forest with accuracy > 85%

3. Architecture

Frontend:

Developed using HTML served via Flask's `render_template()`. Responsive form-based interface for capturing patient data.

Backend:

Python Flask handles routing, form submission, and model integration. Processes user input, invokes the trained model, and returns prediction results.

Database:

The dataset contains patient medical records with features like age, alcohol use, cholesterol levels, and liver function indicators. The target variable is used to predict liver cirrhosis using machine learning.

4. Setup Instructions

Prerequisites:

- Python 3.11+
- pip
- Flask
- scikit-learn, numpy, pandas, joblib

Installation:

```
# Clone the repo cd
Project
pip install -r requirements.txt # If using requirements file python
app.py
```

5. Folder Structure

```
Project/
├── Document
├── app.py
├── liver_cirrhosis_model.pkl
├── static/
│   └── (CSS, JS, images if any)
├── templates/
│   ├── index.html
│   └── result.html
├── HealthCareData.csv
└── liver_model_training.ipynb
```

6. Running the Application

```
# Start Flask backend python
app.py
```

```
# Visit in browser
http://127.0.0.1:5000/
```

7. API Documentation

- **POST /predict**
 - Description: Takes user input and returns prediction result
 - Parameters: Age, Alcohol Consumption, Blood Parameters (as form data)
 - Response: Prediction result (e.g., "Likely to have cirrhosis")

8. Authentication

- Not implemented in current version.
- Can be added using Flask-Login and JWT for user sessions

9. User Interface

- **index.html:** Data input form with fields for age, blood values, etc.
- **result.html:** Displays whether the patient is likely or unlikely to have cirrhosis.

10. Testing

- Manual Testing: Tested with multiple patient records for accuracy.
- Model Testing: Evaluated using accuracy, precision, recall, and F1-score on test dataset

```
# Predictions
Y_pred = model.predict(X_test)

# Evaluation
print("✅ Model Evaluation:\n")
print("Accuracy      :", round(accuracy_score(Y_test, Y_pred) * 100, 2), "%")
print("Precision     :", round(precision_score(Y_test, Y_pred) * 100, 2), "%")
print("Recall        :", round(recall_score(Y_test, Y_pred) * 100, 2), "%")
print("F1 Score      :", round(f1_score(Y_test, Y_pred) * 100, 2), "%")

print("\nConfusion Matrix:\n", confusion_matrix(Y_test, Y_pred))
print("\nClassification Report:\n", classification_report(Y_test, Y_pred))
```

✅ Model Evaluation:

```
Accuracy      : 100.0 %
Precision     : 100.0 %
Recall        : 100.0 %
F1 Score      : 100.0 %
```

Confusion Matrix:

```
[[187  0]
 [ 0  3]]
```

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	187
1	1.00	1.00	1.00	3
accuracy			1.00	190
macro avg	1.00	1.00	1.00	190
weighted avg	1.00	1.00	1.00	190

11. Screenshots or Demo

- Screenshot of Input Form

Liver Cirrhosis Prediction

Enter Patient Details

Age:

Quantity of Alcohol Consumption:

Diabetes Result (yes/no):

Blood Pressure (systolic/diastolic like 120/80):

Blood Test Results

Hemoglobin:

PCV:

Polymorphs:

Lymphocytes:

Platelet Count:

Indirect Bilirubin:

Total Protein:

Albumin:

Globulin:

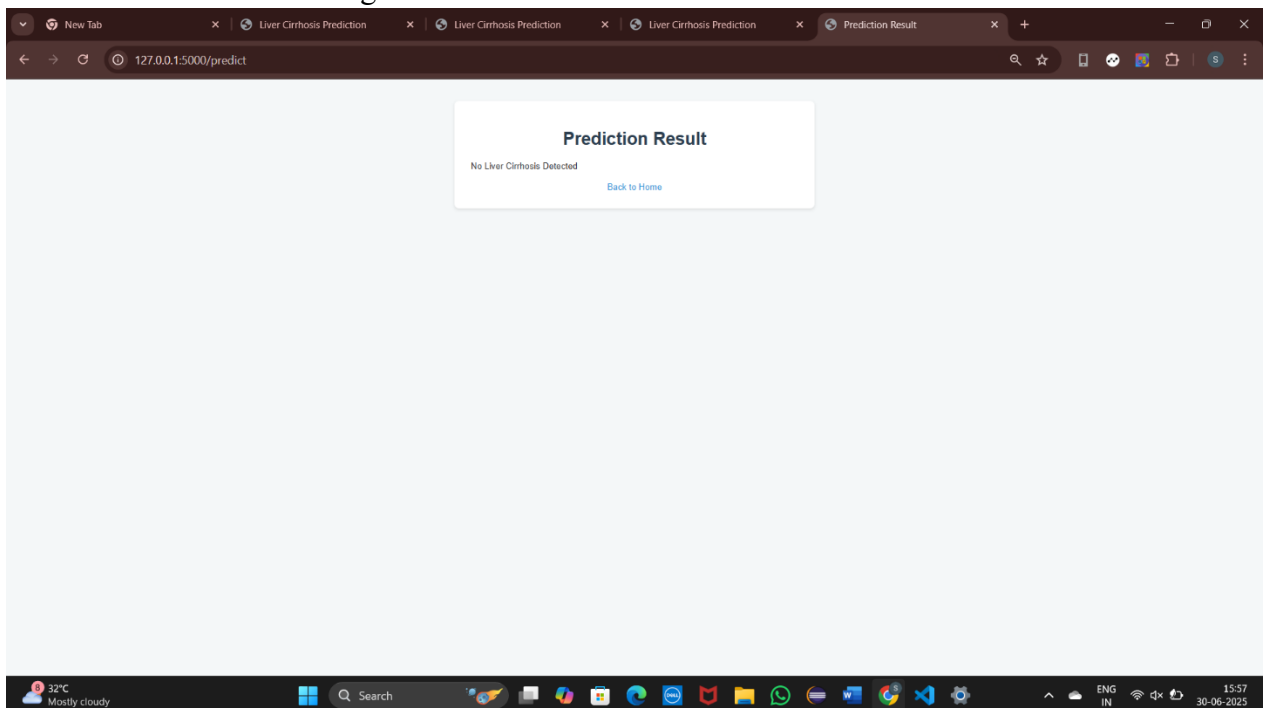
A/G Ratio:

AL Phosphatase:

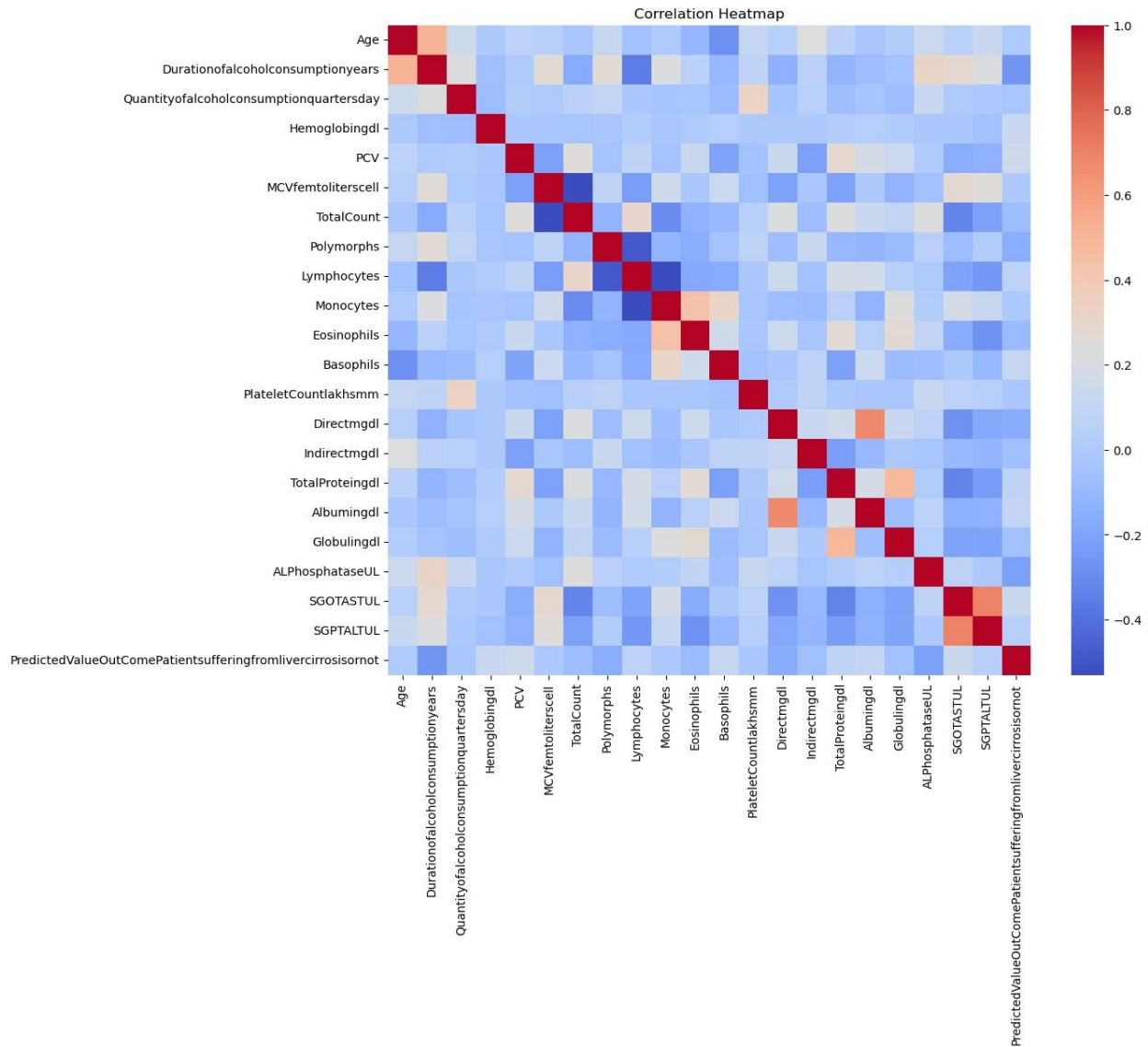
USG Abdomen (yes/no):

[Predict](#)

- Screenshot of Result Page



- Correlation Heatmap from Training Notebook



12. Known Issues

- Does not handle missing inputs dynamically on frontend.
- No persistent storage of results.

13. Future Enhancements

- Add login and user dashboard
- Save prediction history to database (e.g., MongoDB)
- Host on cloud (Heroku, AWS, etc.)
- Deploy RESTful API for mobile integration .