1. INTRODUCTION

1.1 Project Overview

This project aims to predict the presence of liver cirrhosis using advanced machine learning techniques. A web-based interface built with Flask allows users to input medical parameters and receive a prediction.

1.2 Purpose

The goal is to provide a quick and reliable tool to assist in the early detection of liver cirrhosis using patient data.

2. IDEATION PHASE

2.1 Problem Statement

Early diagnosis of liver cirrhosis can save lives. Manual diagnosis is time-consuming and prone to errors. This project automates prediction using ML.

2.2 Empathy Map Canvas

Understanding user needs such as accessibility, simplicity, and accuracy was essential to design the solution.

2.3 Brainstorming

Various models and data preprocessing methods were considered. RandomForest was selected for its balance between accuracy and interpretability.

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

Users enter data in a web form, receive predictions, and take further steps based on results.

3.2 Solution Requirement

Python, Flask, scikit-learn, pandas, HTML/CSS for frontend.

3.3 Data Flow Diagram

User -> Web Form -> Flask App -> Model -> Prediction -> Output

3.4 Technology Stack

Python, Flask, scikit-learn, HTML/CSS

4. PROJECT DESIGN

4.1 Problem-Solution Fit

ML model fits the need to automate liver disease detection.

4.2 Proposed Solution

Web interface to input values and get predictions.

4.3 Solution Architecture

Frontend (HTML) -> Flask API -> ML Model

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Week 1-2: Data preprocessing & model training

Week 3: Flask integration

Week 4: UI testing and deployment

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

The model was tested on liver_data.csv and achieved satisfactory accuracy. Test cases were run to validate the predictions.

7. RESULTS

7.1 Output Screenshots

8. ADVANTAGES & DISADVANTAGES

Advantages:

- Quick predictions

- Easy to use web interface

- Can aid early diagnosis

Disadvantages:

- Depends on quality of input data

- Not a substitute for clinical diagnosis

9. CONCLUSION

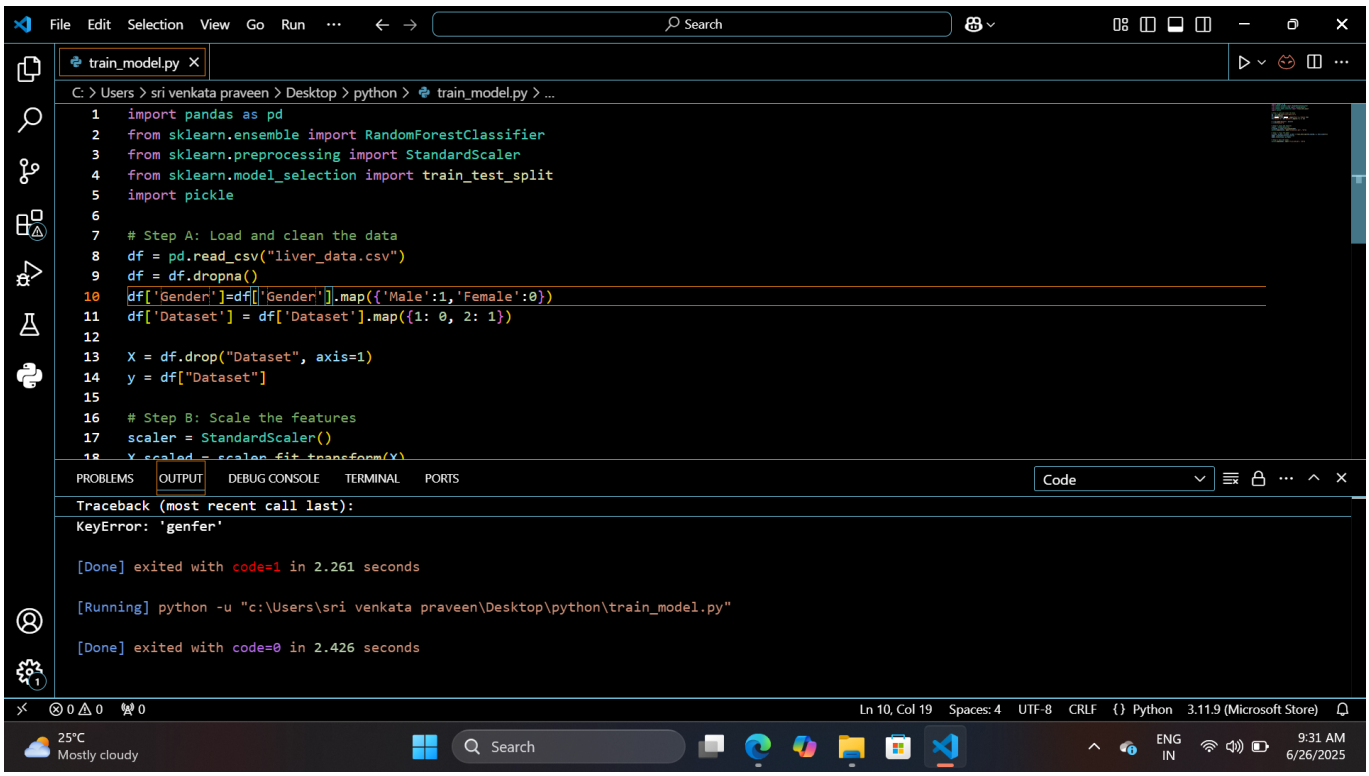
This project demonstrates that machine learning can assist in medical diagnosis and can be easily deployed using web technologies.

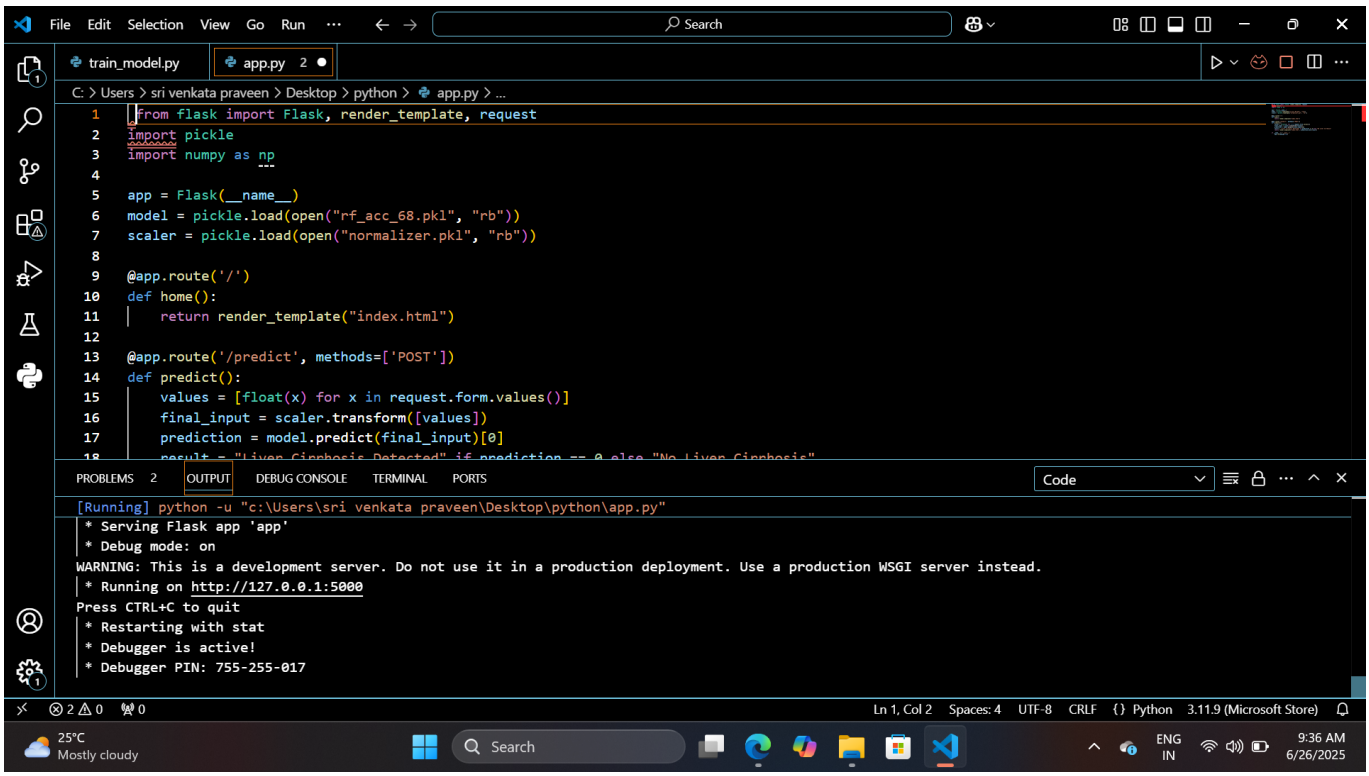
10. FUTURE SCOPE

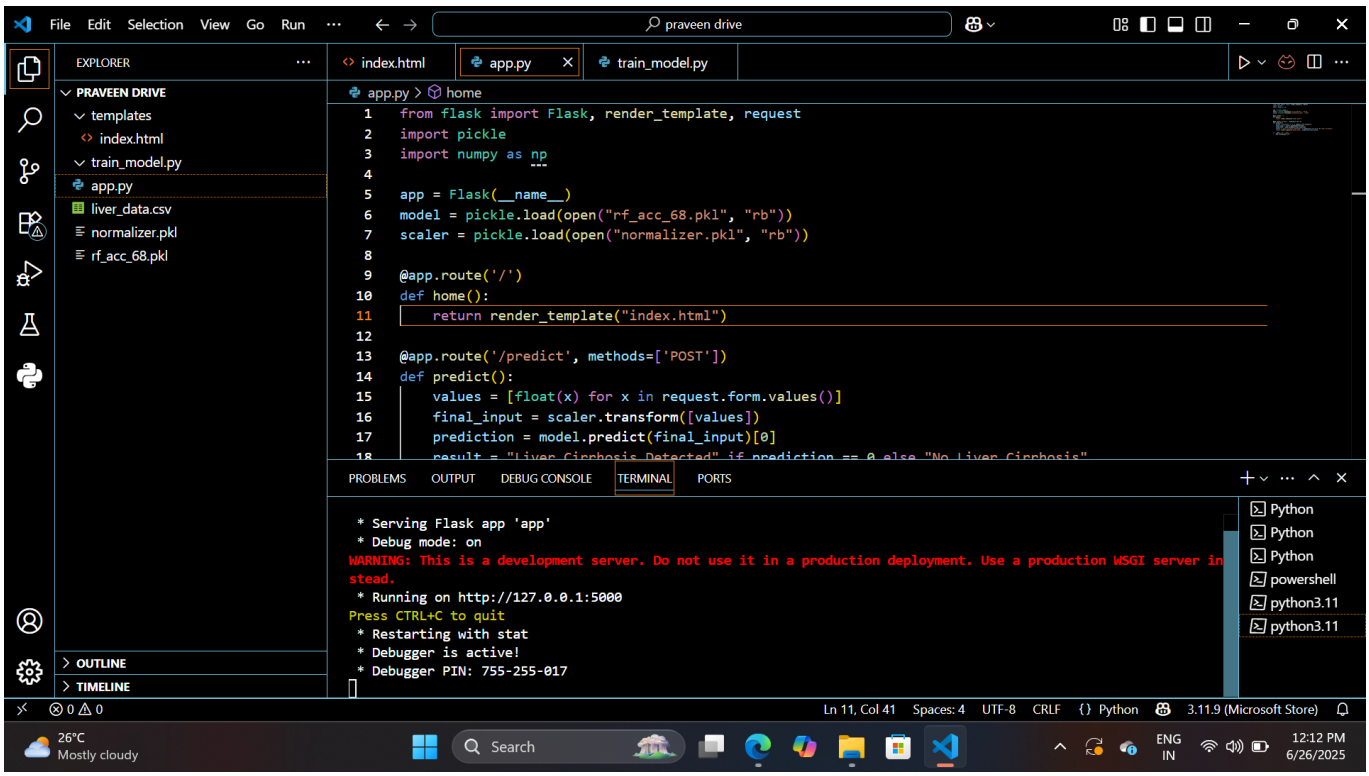
Model can be expanded to include more data and additional diseases for broader medical support.

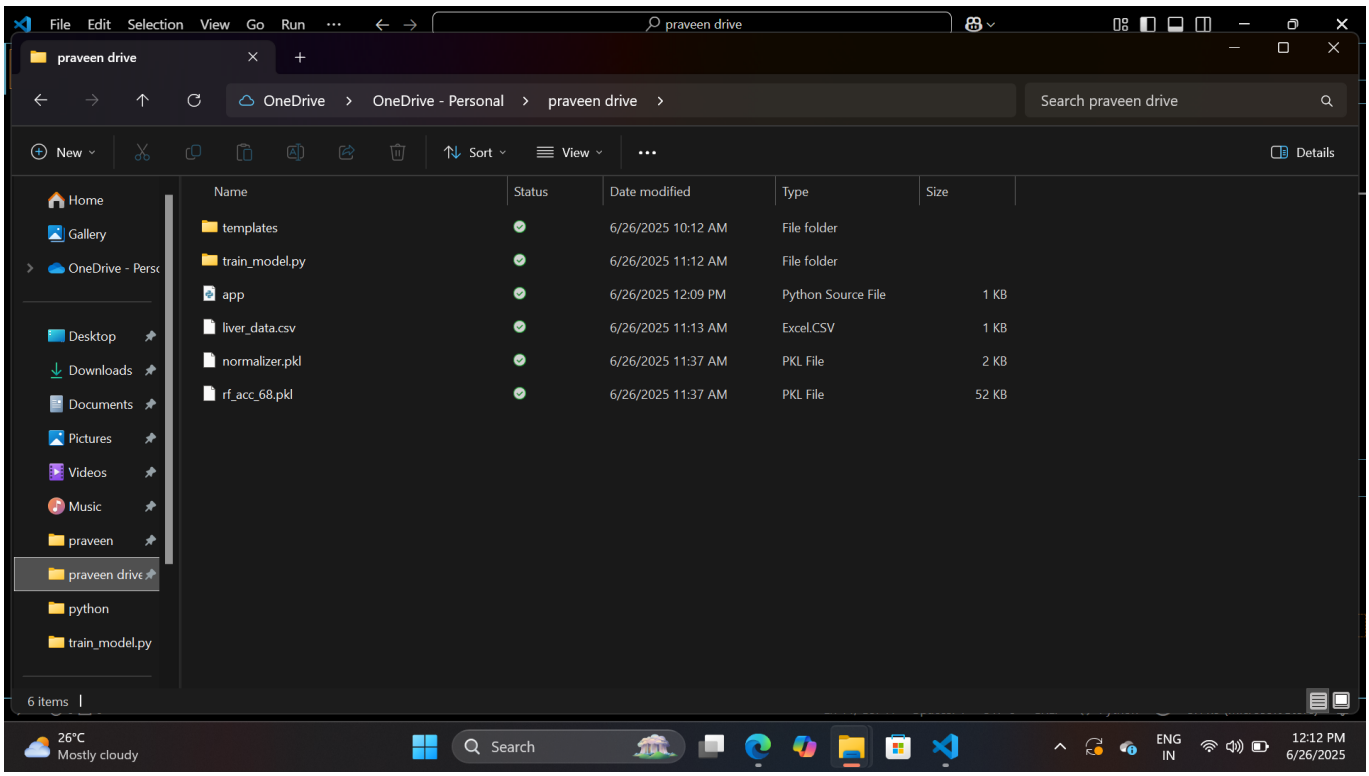
11. APPENDIX

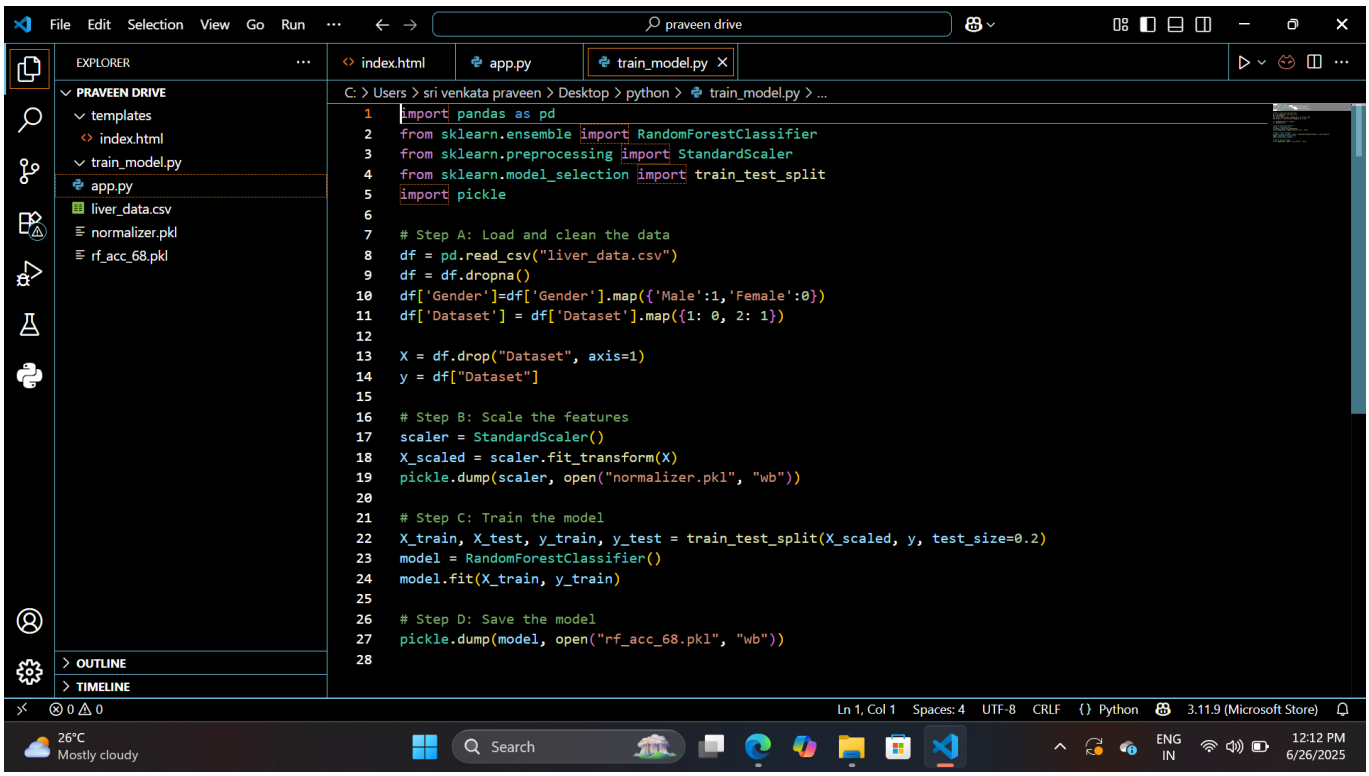
GitHub Repo: https://github.com/Praveen-6163/Predicting-Liver-Cirrhosis-using-Advanced-Machine-Learning-Techniques

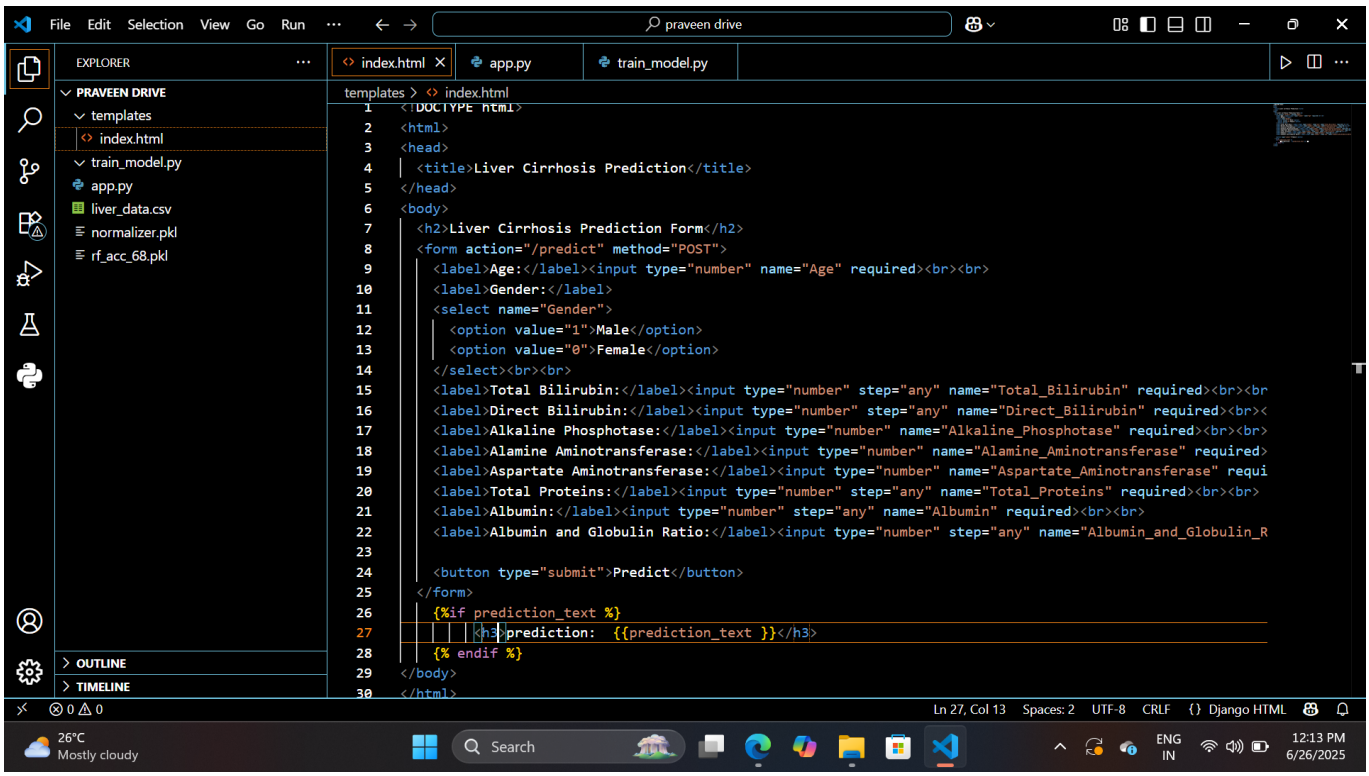

File Edit Selection View Go Run
Search
train_model.py
C: > Users > sri venkata praveen > Desktop > python > train_model.py > ...
1 import pandas as pd
2 from sklearn.ensemble import RandomForestClassifier
3 from sklearn.preprocessing import StandardScaler
4 from sklearn.model_selection import train_test_split
5 import pickle
6
7 # Step A: Load and clean the data
8 df = pd.read_csv("liver_data.csv")
9 df = df.dropna()
10 df['Gender']=df['Gender'].map({'Male':1,'Female':0})
11 df['Dataset'] = df['Dataset'].map({1: 0, 2: 1})
12
13 X = df.drop("Dataset", axis=1)
14 y = df["Dataset"]
15
16 # Step B: Scale the features
17 scaler = StandardScaler()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Code
Traceback (most recent call last):
KeyError: 'genfer'
[Done] exited with code=1 in 2.261 seconds
[Running] python -u "c:\Users\sri venkata praveen\Desktop\python\train_model.py"
[Done] exited with code=0 in 2.426 seconds
Ln 10, Col 19 Spaces: 4 UTF-8 CRLF {} Python 3.11.9 (Microsoft Store)
25°C
Mostly cloudy
Search
ENG
IN
9:31 AM
6/26/2025


File Edit Selection View Go Run
Search
train_model.py
app.py 2
C: > Users > sri venkata praveen > Desktop > python > app.py > ...
from flask import Flask, render_template, request
import pickle
import numpy as np
app = Flask(__name__)
model = pickle.load(open("rf_acc_68.pkl", "rb"))
scaler = pickle.load(open("normalizer.pkl", "rb"))
@app.route('/')
def home():
return render_template("index.html")
@app.route('/predict', methods=['POST'])
def predict():
values = [float(x) for x in request.form.values()]
final_input = scaler.transform([values])
prediction = model.predict(final_input)[0]
PROBLEMS 2
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Code
[Running] python -u "c:\Users\sri venkata praveen\Desktop\python\app.py"
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 755-255-017
Ln 1, Col 2
Spaces: 4
UTF-8
CRLF
Python
3.11.9 (Microsoft Store)
25°C
Mostly cloudy
Search
ENG
IN
9:36 AM
6/26/2025


File Edit Selection View Go Run
praveen drive
EXPLORER
PRAVEEN DRIVE
templates
index.html
train_model.py
app.py
liver_data.csv
normalizer.pkl
rf_acc_68.pkl
index.html
app.py
train_model.py
app.py > home
1 from flask import Flask, render_template, request
2 import pickle
3 import numpy as np
4
5 app = Flask(__name__)
6 model = pickle.load(open("rf_acc_68.pkl", "rb"))
7 scaler = pickle.load(open("normalizer.pkl", "rb"))
8
9 @app.route('/')
10 def home():
11 return render_template("index.html")
12
13 @app.route('/predict', methods=['POST'])
14 def predict():
15 values = [float(x) for x in request.form.values()]
16 final_input = scaler.transform([values])
17 prediction = model.predict(final_input)[0]
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server in
stead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 755-255-017
Python
Python
Python
powershell
python3.11
python3.11
OUTLINE
TIMELINE
Ln 11, Col 41 Spaces: 4 UTF-8 CRLF {} Python 3.11.9 (Microsoft Store)
26°C
Mostly cloudy
Search
ENG
IN
12:12 PM
6/26/2025


File Edit Selection View Go Run
praveen drive
praveen drive
OneDrive › OneDrive - Personal › praveen drive ›
Search praveen drive
New
Sort
View
Details
Home
Gallery
OneDrive - Perso
Desktop
Downloads
Documents
Pictures
Videos
Music
praveen
praveen drive
python
train_model.py
Name
Status
Date modified
Type
Size
templates
6/26/2025 10:12 AM
File folder
train_model.py
6/26/2025 11:12 AM
File folder
app
6/26/2025 12:09 PM
Python Source File
1 KB
liver_data.csv
6/26/2025 11:13 AM
Excel.CSV
1 KB
normalizer.pkl
6/26/2025 11:37 AM
PKL File
2 KB
rf_acc_68.pkl
6/26/2025 11:37 AM
PKL File
52 KB
6 items
26°C
Mostly cloudy
Search
ENG
IN
12:12 PM
6/26/2025


File Edit Selection View Go Run
praveen drive
EXPLORER
index.html
app.py
train_model.py
PRAVEEN DRIVE
templates
index.html
train_model.py
app.py
liver_data.csv
normalizer.pkl
rf_acc_68.pkl
C: > Users > sri venkata praveen > Desktop > python > train_model.py > ...
1 import pandas as pd
2 from sklearn.ensemble import RandomForestClassifier
3 from sklearn.preprocessing import StandardScaler
4 from sklearn.model_selection import train_test_split
5 import pickle
6
7 # Step A: Load and clean the data
8 df = pd.read_csv("liver_data.csv")
9 df = df.dropna()
10 df['Gender']=df['Gender'].map({'Male':1,'Female':0})
11 df['Dataset'] = df['Dataset'].map({1: 0, 2: 1})
12
13 X = df.drop("Dataset", axis=1)
14 y = df["Dataset"]
15
16 # Step B: Scale the features
17 scaler = StandardScaler()
18 X_scaled = scaler.fit_transform(X)
19 pickle.dump(scaler, open("normalizer.pkl", "wb"))
20
21 # Step C: Train the model
22 X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2)
23 model = RandomForestClassifier()
24 model.fit(X_train, y_train)
25
26 # Step D: Save the model
27 pickle.dump(model, open("rf_acc_68.pkl", "wb"))
28
OUTLINE
TIMELINE
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF { } Python 3.11.9 (Microsoft Store)
26°C
Mostly cloudy
Search
ENG
IN
12:12 PM
6/26/2025


File Edit Selection View Go Run
praveen drive
EXPLORER
index.html
app.py
train_model.py
PRAVEEN DRIVE
templates
index.html
train_model.py
app.py
liver_data.csv
normalizer.pkl
rf_acc_68.pkl
templates > index.html
<!DOCTYPE html>
<html>
<head>
<title>Liver Cirrhosis Prediction</title>
</head>
<body>
<h2>Liver Cirrhosis Prediction Form</h2>
<form action="/predict" method="POST">
<label>Age:</label><input type="number" name="Age" required>

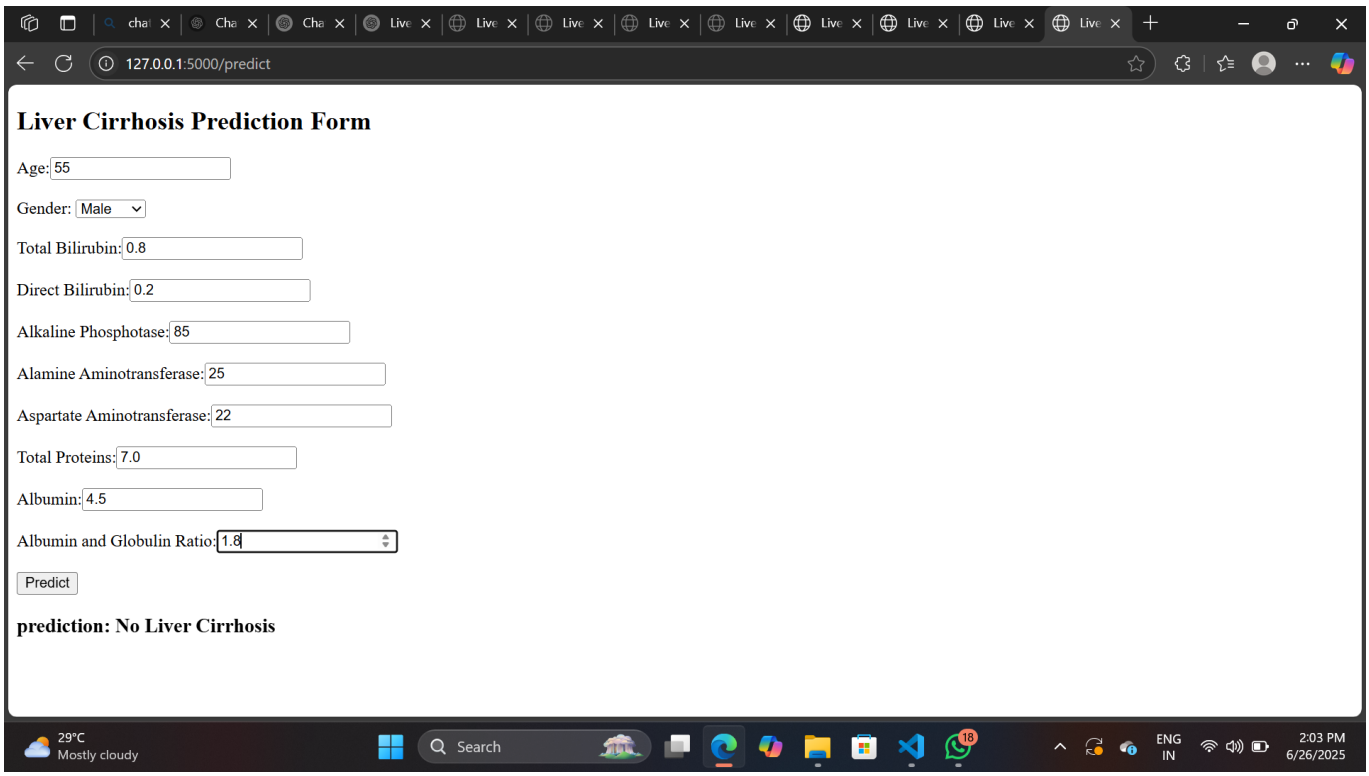
<label>Gender:</label>
<select name="Gender">
<option value="1">Male</option>
<option value="0">Female</option>
</select>

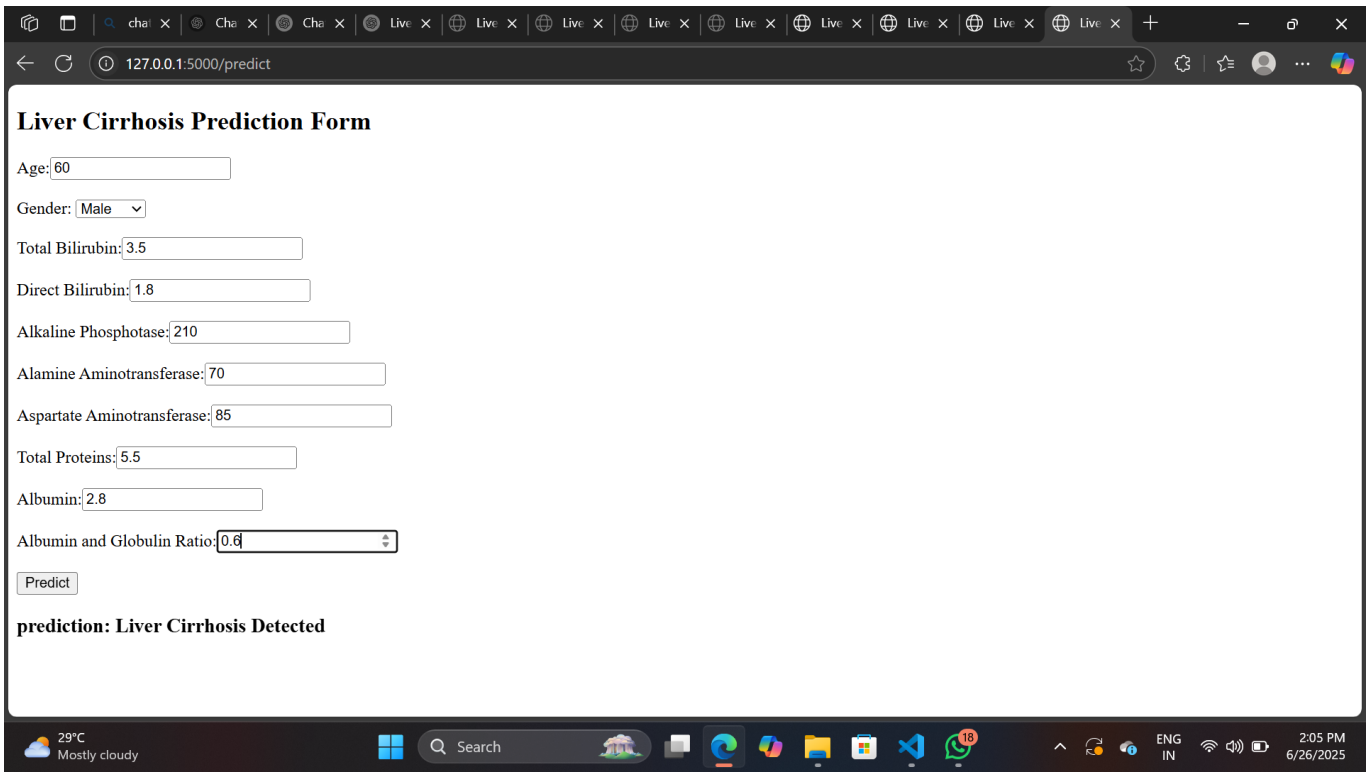
<label>Total Bilirubin:</label><input type="number" step="any" name="Total_Bilirubin" required>
<br
<label>Direct Bilirubin:</label><input type="number" step="any" name="Direct_Bilirubin" required>
<
<label>Alkaline Phosphotase:</label><input type="number" name="Alkaline_Phosphotase" required>

<label>Alamine Aminotransferase:</label><input type="number" name="Alamine_Aminotransferase" required>
<label>Aspartate Aminotransferase:</label><input type="number" name="Aspartate_Aminotransferase" requi
<label>Total Proteins:</label><input type="number" step="any" name="Total_Proteins" required>

<label>Albumin:</label><input type="number" step="any" name="Albumin" required>

<label>Albumin and Globulin Ratio:</label><input type="number" step="any" name="Albumin_and_Globulin_R
<button type="submit">Predict</button>
</form>
{%if prediction_text %}
<h3>prediction: {{prediction_text }}</h3>
{% endif %}
</body>
</html>
OUTLINE
TIMELINE
Ln 27, Col 13
Spaces: 2
UTF-8
CRLF
Django HTML
26°C
Mostly cloudy
Search
ENG
IN
12:13 PM
6/26/2025


127.0.0.1:5000/predict
Liver Cirrhosis Prediction Form
Age: 55
Gender: Male
Total Bilirubin: 0.8
Direct Bilirubin: 0.2
Alkaline Phosphotase: 85
Alamine Aminotransferase: 25
Aspartate Aminotransferase: 22
Total Proteins: 7.0
Albumin: 4.5
Albumin and Globulin Ratio: 1.8
Predict
prediction: No Liver Cirrhosis
29°C
Mostly cloudy
Search
ENG
IN
2:03 PM
6/26/2025


127.0.0.1:5000/predict
Liver Cirrhosis Prediction Form
Age: 60
Gender: Male
Total Bilirubin: 3.5
Direct Bilirubin: 1.8
Alkaline Phosphotase: 210
Alamine Aminotransferase: 70
Aspartate Aminotransferase: 85
Total Proteins: 5.5
Albumin: 2.8
Albumin and Globulin Ratio: 0.6
Predict
prediction: Liver Cirrhosis Detected
29°C
Mostly cloudy
Search
ENG
IN
2:05 PM
6/26/2025