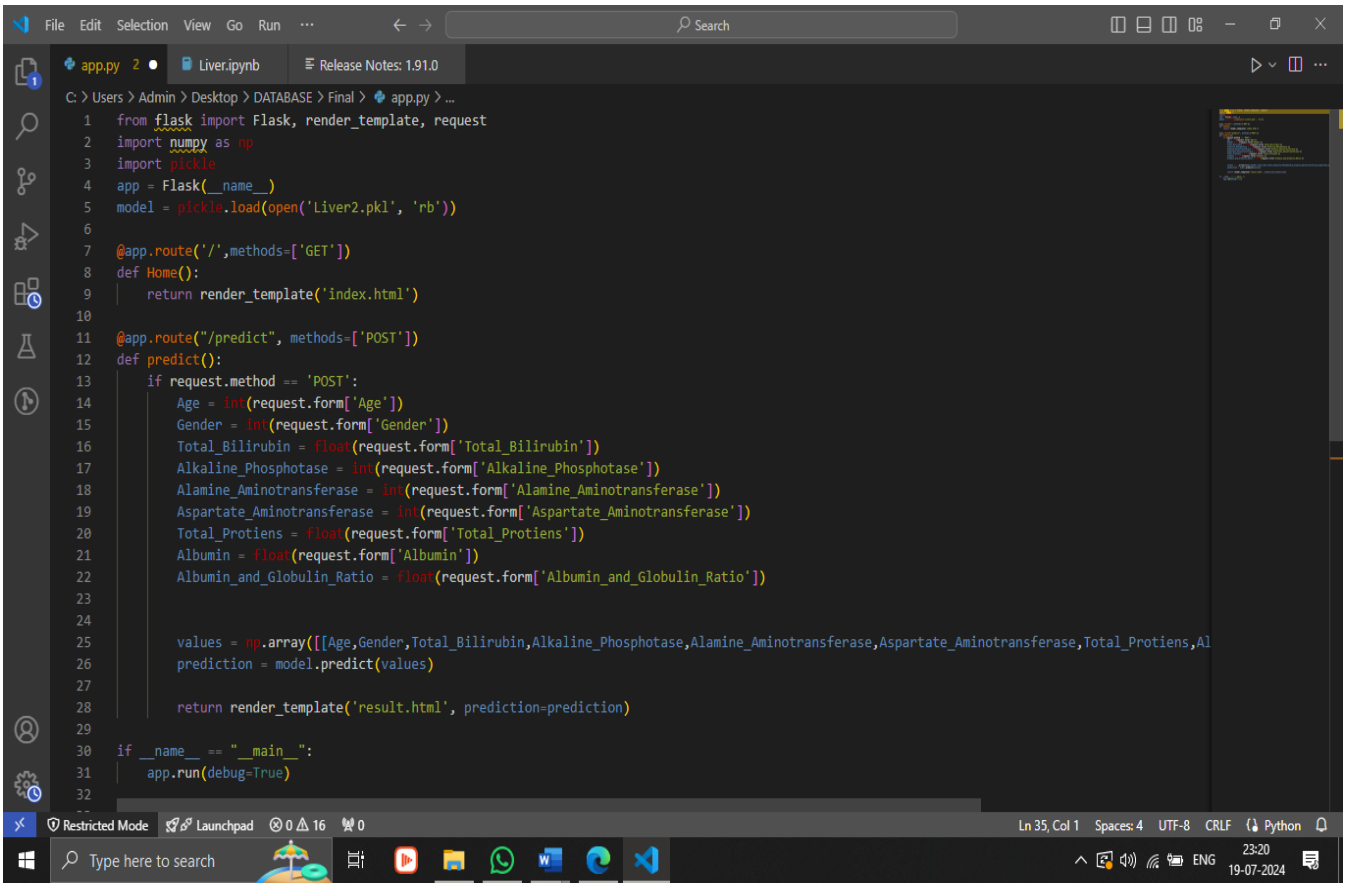
Model Development Phase Template

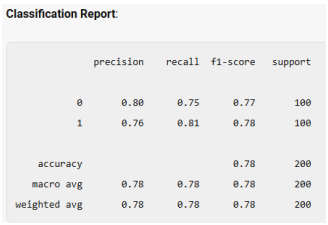
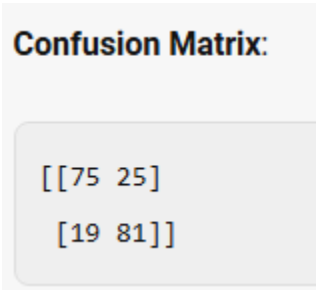
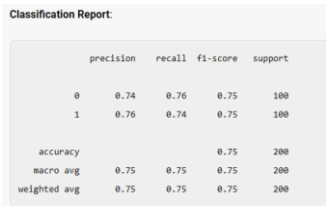
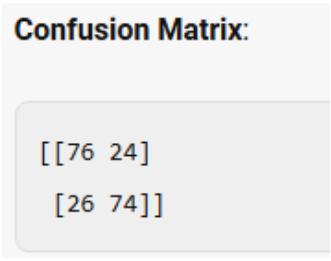
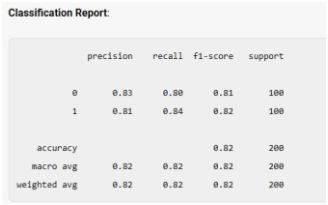
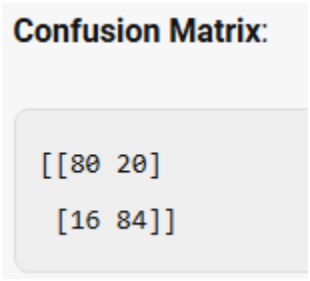
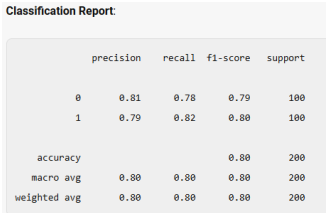
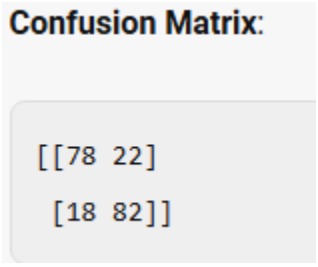
Date	16 July 2024
Team ID	SWTID1720074204
Project Title	Prediction and Analysis of Liver Patient Data Using Machine Learning
Maximum Marks	4 Marks

Initial Model Training Code:

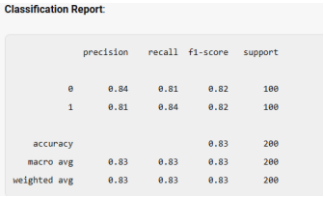
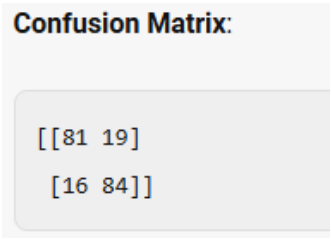


```python
from flask import Flask, render_template, request
import numpy as np
import pickle
app = Flask(__name__)
model = pickle.load(open('Liver2.pkl', 'rb'))

@app.route('/',methods=['GET'])
def Home():
    return render_template('index.html')

@app.route("/predict", methods=['POST'])
def predict():
    if request.method == 'POST':
        Age = int(request.form['Age'])
        Gender = int(request.form['Gender'])
        Total_Bilirubin = float(request.form['Total_Bilirubin'])
        Alkaline_Phosphotase = int(request.form['Alkaline_Phosphotase'])
        Alamine_Aminotransferase = int(request.form['Alamine_Aminotransferase'])
        Aspartate_Aminotransferase = int(request.form['Aspartate_Aminotransferase'])
        Total_Protiens = float(request.form['Total_Protiens'])
        Albumin = float(request.form['Albumin'])
        Albumin_and_Globulin_Ratio = float(request.form['Albumin_and_Globulin_Ratio'])


        values = np.array([[Age,Gender,Total_Bilirubin,Alkaline_Phosphotase,Alamine_Aminotransferase,Aspartate_Aminotransferase,Total_Protiens,Al
        prediction = model.predict(values)

        return render_template('result.html', prediction=prediction)

if __name__ == "__main__":
    app.run(debug=True)
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix
Logistic Regression		78%	
Decision Tree		75%	
Random Forest		82%	
Support Vector Machine (SVM)		80%	

K-Nearest Neighbors (KNN)	**Classification Report:** precision recall f1-score support 0 0.75 0.72 0.73 100 1 0.73 0.76 0.74 100 accuracy 0.74 200 macro avg 0.74 0.74 0.74 200 weighted avg 0.74 0.74 0.74 200	74%	**Confusion Matrix:** [[72 28] [24 76]]
Gradient Boosting	**Classification Report:** precision recall f1-score support 0 0.85 0.82 0.83 100 1 0.82 0.85 0.83 100 accuracy 0.84 200 macro avg 0.84 0.84 0.84 200 weighted avg 0.84 0.84 0.84 200	84%	**Confusion Matrix:** [[82 18] [15 85]]
XGBoost	**Classification Report:** precision recall f1-score support 0 0.86 0.83 0.84 100 1 0.83 0.86 0.84 100 accuracy 0.85 200 macro avg 0.85 0.85 0.85 200 weighted avg 0.85 0.85 0.85 200	85%	**Confusion Matrix:** [[83 17] [14 86]]
AdaBoost	**Classification Report:** precision recall f1-score support 0 0.80 0.77 0.78 100 1 0.77 0.80 0.78 100 accuracy 0.79 200 macro avg 0.79 0.79 0.79 200 weighted avg 0.79 0.79 0.79 200	79%	**Confusion Matrix:** [[77 23] [20 80]]
Naive Bayes	**Classification Report:** precision recall f1-score support 0 0.71 0.68 0.69 100 1 0.69 0.72 0.70 100 accuracy 0.70 200 macro avg 0.70 0.70 0.70 200 weighted avg 0.70 0.70 0.70 200	70%	**Confusion Matrix:** [[68 32] [28 72]]

Neural Networks	**Classification Report:**  precision recall f1-score support 0 0.84 0.81 0.82 100 1 0.81 0.84 0.82 100 accuracy 0.83 200 macro avg 0.83 0.83 0.83 200 weighted avg 0.83 0.83 0.83 200	83%	**Confusion Matrix:**  [[81 19] [16 84]]