```
# liver_care_predictor.py

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report

# Load sample liver patient dataset
def load_data():
    # Sample dummy data (in practice, load from CSV or hospital DB)
    data = {
        'Age': [65, 62, 45, 38, 50],
        'Total_Bilirubin': [0.7, 1.0, 1.5, 0.9, 1.2],
        'Alkaline_Phosphotase': [187, 202, 290, 180, 250],
        'SGPT': [16, 27, 35, 20, 30],
        'SGOT': [18, 30, 45, 25, 40],
        'Albumin': [4.0, 3.5, 3.0, 4.2, 3.8],
        'A_G_Ratio': [1.1, 0.9, 0.7, 1.3, 1.0],
        'Liver_Disease': [1, 1, 1, 0, 0]  # 1 = Has disease, 0 = Healthy
    }
    return pd.DataFrame(data)

# Train model
def train_model(df):
    X = df.drop('Liver_Disease', axis=1)
    y = df['Liver_Disease']

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    model = DecisionTreeClassifier()
    model.fit(X_train, y_train)

    predictions = model.predict(X_test)
    print("Model Evaluation:\n", classification_report(y_test, predictions))

    return model

# Predict liver disease from user input
def predict_user(model):
    print("\n--- Liver Disease Risk Predictor ---")
    age = float(input("Enter Age: "))
    tb = float(input("Enter Total Bilirubin: "))
    ap = float(input("Enter Alkaline Phosphotase: "))
    sgpt = float(input("Enter SGPT: "))
    sgot = float(input("Enter SGOT: "))
    albumin = float(input("Enter Albumin: "))
    ag_ratio = float(input("Enter Albumin and Globulin Ratio: "))

    sample = pd.DataFrame([[age, tb, ap, sgpt, sgot, albumin, ag_ratio]],
                          columns=['Age', 'Total_Bilirubin', 'Alkaline_Phosphotase', 'SGPT', '

    result = model.predict(sample)[0]
    print("\nResult: ", "Likely Liver Disease" if result == 1 else "Likely Healthy")
```

```
# Main
if __name__ == "__main__":
    df = load_data()
    model = train_model(df)
    predict_user(model)
```