```
# Import necessary libraries and packages

import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy_score

import streamlit as st


# Load the dataset

data = pd.read_csv("liver_patient_data.csv")


# Preprocess the data

# Drop rows with missing values

data.dropna(inplace=True)


# Split the data into features and target

X = data.drop("Liver_patient", axis=1)

y = data["Liver_patient"]


# Split the data into training and testing sets

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


# Create a random forest classifier

rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
```

```
# Train the classifier

rf_classifier.fit(X_train, y_train)


# Make predictions on the testing set

y_pred = rf_classifier.predict(X_test)


# Evaluate the performance of the classifier

accuracy = accuracy_score(y_test, y_pred)


# Create a Streamlit app to display the results

st.title("Liver Patient Analysis App")

st.write("Accuracy: ", accuracy)


# Create a form for the user to enter patient information

age = st.number_input("Age", min_value=1, max_value=120, step=1)

gender = st.selectbox("Gender", ["Male", "Female"])

total_bilirubin = st.number_input("Total Bilirubin", min_value=0.1, max_value=100.0, step=0.1)

direct_bilirubin = st.number_input("Direct Bilirubin", min_value=0.1, max_value=50.0, step=0.1)

alkaline_phosphotase = st.number_input("Alkaline Phosphotase", min_value=0, max_value=3000,
step=1)

alamine_aminotransferase = st.number_input("Alamine Aminotransferase", min_value=0,
max_value=2000, step=1)

aspartate_aminotransferase = st.number_input("Aspartate Aminotransferase", min_value=0,
max_value=2000, step=1)

total_protein = st.number_input("Total Protein", min_value=0.1, max_value=10.0, step=0.1)

albumin = st.number_input("Albumin", min_value=0.1, max_value=10.0, step=0.1)
```

```
albumin_and_globulin_ratio = st.number_input("Albumin and Globulin Ratio", min_value=0.1,
max_value=10.0, step=0.1)


# Create a feature vector from the user input

input_data = np.array([[age, gender, total_bilirubin, direct_bilirubin, alkaline_phosphotase,

                        alamine_aminotransferase, aspartate_aminotransferase, total_protein, albumin,

                        albumin_and_globulin_ratio]])


# Make a prediction on the user input

prediction = rf_classifier.predict(input_data)


# Display the prediction to the user

if prediction == 1:

    st.write("This patient is likely to have liver disease.")

else:

    st.write("This patient is unlikely to have liver disease.")
```