```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix


df=pd.read_csv('/content/heart_disease_uci.csv')


df.info()


df=df.drop(['id','dataset','ca','thal'], axis=1)


print(df.isnull().sum())


for column in df.columns:
    if df[column].dtype in ['float64', 'int64']:  # Numerical columns
        mean_value = df[column].mean()
        df[column].fillna(mean_value, inplace=True)
    else:  # Categorical columns
        mode_value = df[column].mode()[0]
        df[column].fillna(mode_value, inplace=True)


df.info()


from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
for col in df.columns:
    df[col] = le.fit_transform(df[col])
df


import seaborn as sns
correlation_matrix = df.corr()
# Print the correlation matrix
plt.figure(figsize=(14, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
plt.title('Correlation Matrix')
plt.show()


# Separate features (X) and target variable (y)
X = df.drop('num', axis=1)
y = df['num'].astype(int)  # Ensure target is of integer type


# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=4
```