In [1]:

```
#Assignment 2 Group B
#Name:Samiksha Bandgar
#Roll No:3307
#Subject: DSBDAL
#Batch:A


#Perform the following operations using Python on the Heart Diseases data sets
#a.Data Cleaning
#b.Data integration
#c.Data transformation
#d.Error correcting
#e.Data model building
```

In [2]:

```
import pandas as pd
import numpy as np
```

In [3]:

```
df=pd.read_csv(r"C:\Users\Samiksha Bandgar\OneDrive\Desktop\heart.csv")
```

In [4]:

```
df
```

Out[4]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	t
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	

1025 rows × 14 columns

In [5]:

```
df.columns
```

Out[5]:

```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalac
h',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')
```

In [6]:

```
df.head()
```

Out[6]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	targe
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	

In [7]:

```
df.shape
```

Out[7]:

```
(1025, 14)
```

In [8]:

```
df.dtypes
```

Out[8]:

```
age           int64
sex           int64
cp            int64
trestbps      int64
chol          int64
fbs           int64
restecg       int64
thalach       int64
exang         int64
oldpeak     float64
slope         int64
ca            int64
thal          int64
target        int64
dtype: object
```

In [9]:

```
df.describe()
```

Out[9]:

	age	sex	cp	trestbps	chol	fbs	res
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.00000	1025.000000	1025.00
mean	54.434146	0.695610	0.942439	131.611707	246.00000	0.149268	0.52
std	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527	0.52
min	29.000000	0.000000	0.000000	94.000000	126.00000	0.000000	0.00
25%	48.000000	0.000000	0.000000	120.000000	211.00000	0.000000	0.00
50%	56.000000	1.000000	1.000000	130.000000	240.00000	0.000000	1.00
75%	61.000000	1.000000	2.000000	140.000000	275.00000	0.000000	1.00
max	77.000000	1.000000	3.000000	200.000000	564.00000	1.000000	2.00

In [10]:

```
#Data Cleaning
df.isna().sum()
```

Out[10]:

```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

In [11]:

```
df.nunique()
```

Out[11]:

```
age          41
sex           2
cp            4
trestbps     49
chol        152
fbs           2
restecg       3
thalach      91
exang         2
oldpeak      40
slope         3
ca            5
thal          4
target        2
dtype: int64
```

In [12]:

```
duplicate=df.duplicated().sum()
if duplicate:
    print("Duplicated row{}".format(duplicate))
else:
    print("No duplicate")

df['ca'].nunique()
```

```
Duplicated row723
```

Out[12]:

```
5
```

In [13]:

```
df['ca'].nunique()
```

Out[13]:

```
5
```

In [14]:

```
#Data Transformation
df['ca']=df['ca'].astype('object')
```

In [15]:

```
df.dtypes
```

Out[15]:

```
age           int64
sex           int64
cp            int64
trestbps      int64
chol          int64
fbs           int64
restecg       int64
thalach       int64
exang         int64
oldpeak     float64
slope         int64
ca           object
thal          int64
target        int64
dtype: object
```

In [16]:

```
print(df[df.duplicated()])
```

```
      age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  
\
15     34    0   1       118   210    0        1      192      0      0.7
31     50    0   1       120   244    0        1      162      0      1.1
43     46    1   0       120   249    0        0      144      0      0.8
55     55    1   0       140   217    0        1      111      1      5.6
61     66    0   2       146   278    0        0      152      0      0.0
...   ...  ...  ..       ...   ...  ...      ...      ...    ...      ...
1020   59    1   1       140   221    0        1      164      1      0.0
1021   60    1   0       125   258    0        0      141      1      2.8
1022   47    1   0       110   275    0        0      118      1      1.0
1023   50    0   0       110   254    0        0      159      0      0.0
1024   54    1   0       120   188    0        1      113      0      1.4

      slope  ca  thal  target
15        2  0     2       1
31        2  0     2       1
43        2  0     3       0
55        0  0     3       0
61        1  1     2       1
...     ... ..   ...     ...
1020      2  0     2       1
1021      1  1     3       0
1022      1  1     2       0
1023      2  0     2       1
1024      1  1     3       0

[723 rows x 14 columns]
```

In [17]:

```
df.isna().sum()
df=df.fillna(df.median())
df.isnull().sum()
```

Out[17]:

```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

In [18]:

```
subsetl=df[df[ 'sex' ]==0]

subsetl.shape
```

Out[18]:

```
(312, 14)
```

In [19]:

```
subset2=df[df[ 'sex' ]==1]
```

In [20]:

```
subset2.shape
```

Out[20]:

```
(713, 14)
```

In [21]:

```
#Data integration
combine =[subsetl,subset2]
result=pd.concat (combine)
result.shape
```

Out[21]:

```
(1025, 14)
```

In [22]:

```
subset3=df[[ 'age', 'sex', 'cp']]
```

In [23]:

```
subset3
```

Out[23]:

```
      age  sex  cp
   0   52    1   0
   1   53    1   0
   2   70    1   0
   3   61    1   0
   4   62    0   0
 ...  ...  ...  ...
1020   59    1   1
1021   60    1   0
1022   47    1   0
1023   50    0   0
1024   54    1   0

1025 rows × 3 columns
```

In [24]:

```
subset4=df[['chol', 'fbs']]
```

In [25]:

```
subset4
```

Out[25]:

```
      chol  fbs
   0   212    0
   1   203    1
   2   174    0
   3   203    0
   4   294    1
 ...   ...  ...
1020   221    0
1021   258    0
1022   275    0
1023   254    0
1024   188    0

1025 rows × 2 columns
```

In [26]:

```
#Data Integration
combinel=[subset3,subset4]
resultl=pd.concat(combinel)
resultl.shape
```

Out[26]:

```
(2050, 5)
```

In [27]:

```
resultl.head()
```

Out[27]:

	age	sex	cp	chol	fbs
0	52.0	1.0	0.0	NaN	NaN
1	53.0	1.0	0.0	NaN	NaN
2	70.0	1.0	0.0	NaN	NaN
3	61.0	1.0	0.0	NaN	NaN
4	62.0	0.0	0.0	NaN	NaN

In [28]:

```
result.head()
```

Out[28]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	targ
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	
5	58	0	0	100	248	0	0	122	0	1.0	1	0	2	
10	71	0	0	112	149	0	1	125	0	1.6	1	0	2	
11	43	0	0	132	341	1	0	136	1	3.0	1	0	3	
12	34	0	1	118	210	0	1	192	0	0.7	2	0	2	

In [29]:

```
#Data model building
from sklearn.model_selection import train_test_split
x=df[['age', 'sex', 'cp']]
y=df[[ 'restecg', 'thal']]
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
```

In [30]:

```
x_train
```

Out[30]:

	age	sex	cp
315	42	1	3
204	66	0	2
363	53	1	2
5	58	0	0
1017	53	1	0
...	...	...	...
835	49	1	2
192	67	0	2
629	65	1	3
559	67	1	0
684	60	1	2

820 rows × 3 columns

In [31]:

```
y_train
```

Out[31]:

	restecg	thal
315	0	2
204	0	2
363	0	2
5	0	2
1017	1	3
...	...	...
835	0	2
192	0	3
629	0	2
559	1	2
684	0	2

820 rows × 2 columns

In [32]:

```
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier(criterion='entropy',max_depth=2)
```

In [33]:

```
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
feature_names=df.columns[0:7]
print(feature_names,end='')
```

```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg'], dtype='o
bject')
```

In [34]:

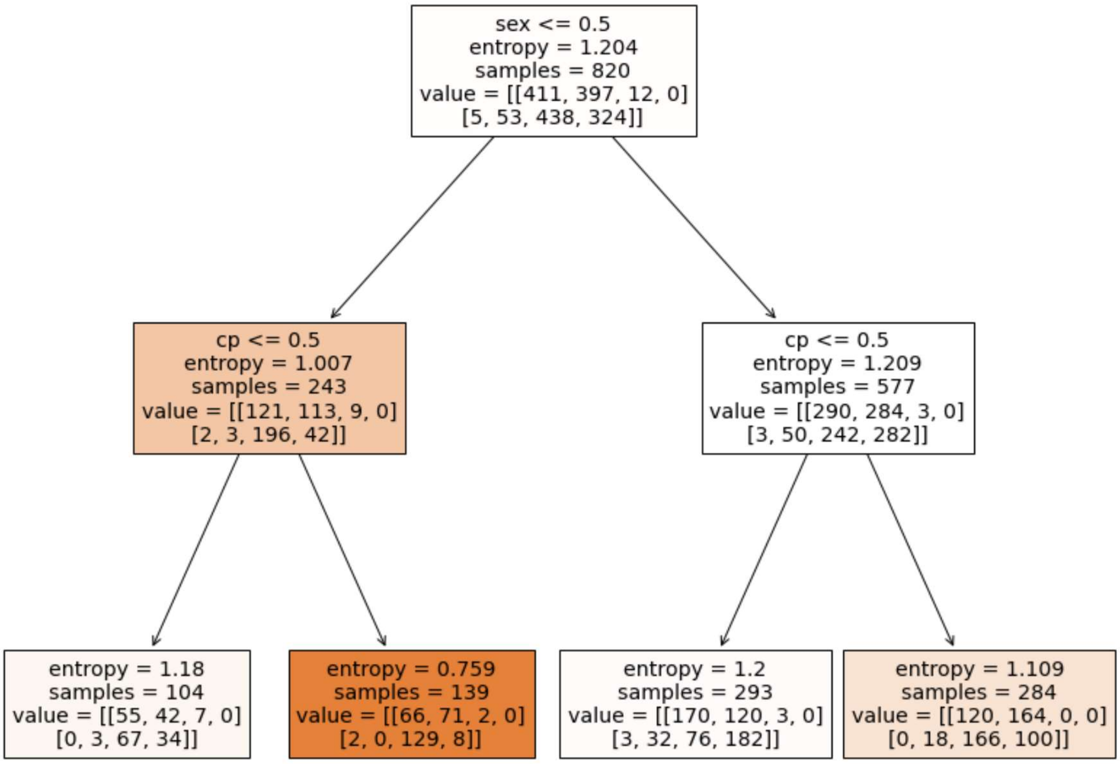
```
class_names=[str(x) for x in model.classes_]
class_names
```

Out[34]:

```
['[0 1 2]', '[0 1 2 3]']
```

In [35]:

```
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
from sklearn.tree import plot_tree
fig=plt.figure(figsize=(14,12))
plot_tree(model,feature_names=feature_names,class_names=class_names,filled=True)
plt.savefig("true visualization.png")
```



In [ ]: