

In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
```

In [2]:

```
df=pd.read_csv(r"C:\Users\DELL\Downloads\loan1 (1).csv")
df
```

Out[2]:

	Home Owner	Marital Status	Annual Income	Defaulted Borrower
0	Yes	Single	125	No
1	No	Married	100	No
2	No	Single	70	No
3	Yes	Married	120	No
4	No	Divorced	95	Yes
5	No	Married	60	No
6	Yes	Divorced	220	No
7	No	Single	85	Yes
8	No	Married	75	No
9	No	Single	90	Yes

In [3]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 4 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Home Owner            10 non-null    object
1   Marital Status        10 non-null    object
2   Annual Income         10 non-null    int64
3   Defaulted Borrower    10 non-null    object
dtypes: int64(1), object(3)
memory usage: 452.0+ bytes
```

In [4]:

```
df['Marital Status'].value_counts()
```

Out[4]:

```
Marital Status
Single      4
Married     4
Divorced    2
Name: count, dtype: int64
```

In [5]:

```
df['Annual Income'].value_counts()
```

Out[5]:

```
Annual Income
125      1
100      1
70       1
120      1
95       1
60       1
220      1
85       1
75       1
90       1
Name: count, dtype: int64
```

In [6]:

```
convert = {"Home Owner":{"Yes":1,"No":0}}
df=df.replace(convert)
print(df)
```

	Home Owner	Marital Status	Annual Income	Defaulted	Borrower
0	1	Single	125		No
1	0	Married	100		No
2	0	Single	70		No
3	1	Married	120		No
4	0	Divorced	95		Yes
5	0	Married	60		No
6	1	Divorced	220		No
7	0	Single	85		Yes
8	0	Married	75		No
9	0	Single	90		Yes

In [7]:

```

convert={"Marital Status":{"Single":1,"Married":2,"Divorced":3}}
df=df.replace(convert)
print(df)

```

	Home Owner	Marital Status	Annual Income	Defaulted Borrower
0	1	1	125	No
1	0	2	100	No
2	0	1	70	No
3	1	2	120	No
4	0	3	95	Yes
5	0	2	60	No
6	1	3	220	No
7	0	1	85	Yes
8	0	2	75	No
9	0	1	90	Yes

In [8]:

```

convert={"Defaulted Borrower":{"No":0,"Yes":1}}
df=df.replace(convert)
print(df)

```

	Home Owner	Marital Status	Annual Income	Defaulted Borrower
0	1	1	125	0
1	0	2	100	0
2	0	1	70	0
3	1	2	120	0
4	0	3	95	1
5	0	2	60	0
6	1	3	220	0
7	0	1	85	1
8	0	2	75	0
9	0	1	90	1

In [9]:

```

x=["Home Owner","Marital Status","Annual Income"]
y=["Yes","No"]
all_inputs=df[x]
all_classes=df["Defaulted Borrower"]

```

In [10]:

```

x_train,x_test,y_train,y_test = train_test_split(all_inputs,all_classes,test_size=0.25)

```

In [11]:

```

clt=DecisionTreeClassifier(random_state=0)

```

In []:

```
clt.fit(x_train,y_train)
```

Out[12]:

```
DecisionTreeClassifier  
DecisionTreeClassifier(random_state=0)
```

In []:

```
score=clt.score(x_test,y_test)  
print(score)
```

Drug Dataset

In []:

```
import numpy as np  
import pandas as pd  
import seaborn as sns  
from sklearn.model_selection import train_test_split  
from sklearn.tree import DecisionTreeClassifier
```

In []:

```
df=pd.read_csv(r"C:\Users\DELL\Downloads\drug200.csv")  
df
```

In []:

```
df.info()
```

In []:

```
df.isnull().sum()
```

In [35]:

```
import matplotlib.pyplot as plt  
import seaborn as sns
```

In [36]:

```
df['Age'].value_counts()
```

Out[36]:

Age

47	8
23	7
28	7
49	7
39	6
32	6
50	5
37	5
58	5
60	5
22	5
34	4
72	4
51	4
42	4
26	4
24	4
74	4
67	4
68	4
61	4
56	4
20	4
36	4
45	4
41	4
31	4
43	4
65	4
57	4
53	3
40	3
70	3
59	3
16	3
38	3
15	3
69	3
35	3
18	3
64	3
52	2
55	2
62	2
19	2
29	2
66	2
73	2
46	2
48	2
54	1
17	1
33	1
63	1
30	1
21	1
25	1

Name: count, dtype: int64

In [37]:

```
s=np.array(df['Age'])
s.sort()
print(s)
```

```
[15 15 15 16 16 16 17 18 18 18 19 19 20 20 20 20 21 22 22 22 22 22 23 23
 23 23 23 23 23 24 24 24 24 25 26 26 26 26 28 28 28 28 28 28 28 29 29 30
 31 31 31 31 32 32 32 32 32 32 33 34 34 34 34 35 35 35 36 36 36 36 37 37
 37 37 37 38 38 38 39 39 39 39 39 39 40 40 40 41 41 41 41 42 42 42 42 43
 43 43 43 45 45 45 45 46 46 47 47 47 47 47 47 47 47 48 48 49 49 49 49 49
 49 49 50 50 50 50 50 51 51 51 51 52 52 53 53 53 54 55 55 56 56 56 56 57
 57 57 57 58 58 58 58 58 59 59 59 60 60 60 60 60 61 61 61 61 62 62 63 64
 64 64 65 65 65 65 66 66 67 67 67 67 68 68 68 68 69 69 69 70 70 70 72 72
 72 72 73 73 74 74 74 74]
```

In [38]:

```
np.mean(df['Age'])
```

Out[38]:

44.315

In [39]:

```
df['BP'].value_counts()
```

Out[39]:

```
BP
2    77
0    64
1    59
Name: count, dtype: int64
```

In [40]:

```
df['Sex'].value_counts()
```

Out[40]:

```
Sex
M    104
F     96
Name: count, dtype: int64
```

In [41]:

```
df['Cholesterol'].value_counts()
```

Out[41]:

```
Cholesterol
1    103
0     97
Name: count, dtype: int64
```

In [42]:

```
df['Drug'].value_counts()
```

Out[42]:

```
Drug
5    91
4    54
1    23
3    16
2    16
Name: count, dtype: int64
```

In [43]:

```
s=pd.crosstab(df['Drug'],df['Sex'])
print(s)
```

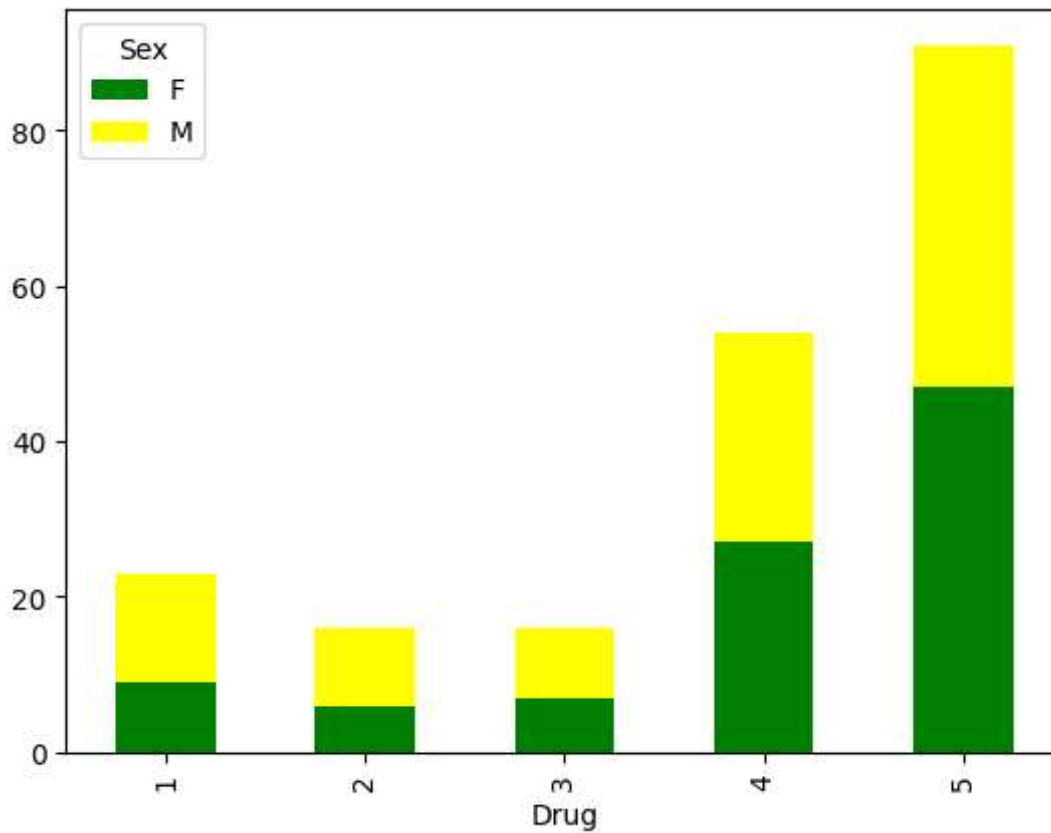
```
Sex    F    M
Drug
1         9   14
2         6   10
3         7    9
4        27   27
5        47   44
```


In [44]:

```
s.plot(kind='bar', stacked=True, color=['green', 'yellow'], grid=False)
```

Out[44]:

<Axes: xlabel='Drug'>



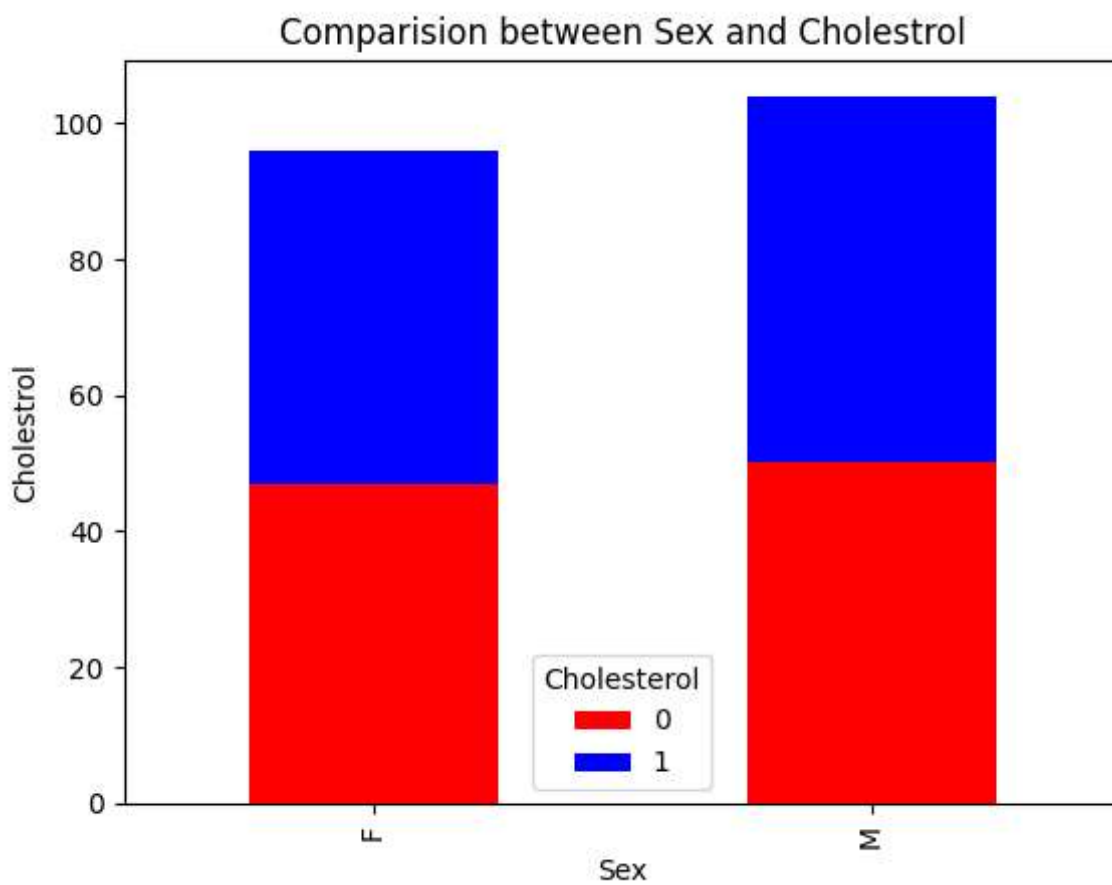
In [45]:

```
s=pd.crosstab(df['Sex'],df['Cholesterol'])  
print(s)  
s.plot(kind='bar',stacked=True,color=['red','blue'],grid=False)  
plt.xlabel('Sex')  
plt.ylabel('Cholesterol')  
plt.title('Comparision between Sex and Cholestrol')
```

Cholesterol	0	1
Sex		
F	47	49
M	50	54

Out[45]:

Text(0.5, 1.0, 'Comparision between Sex and Cholestrol')



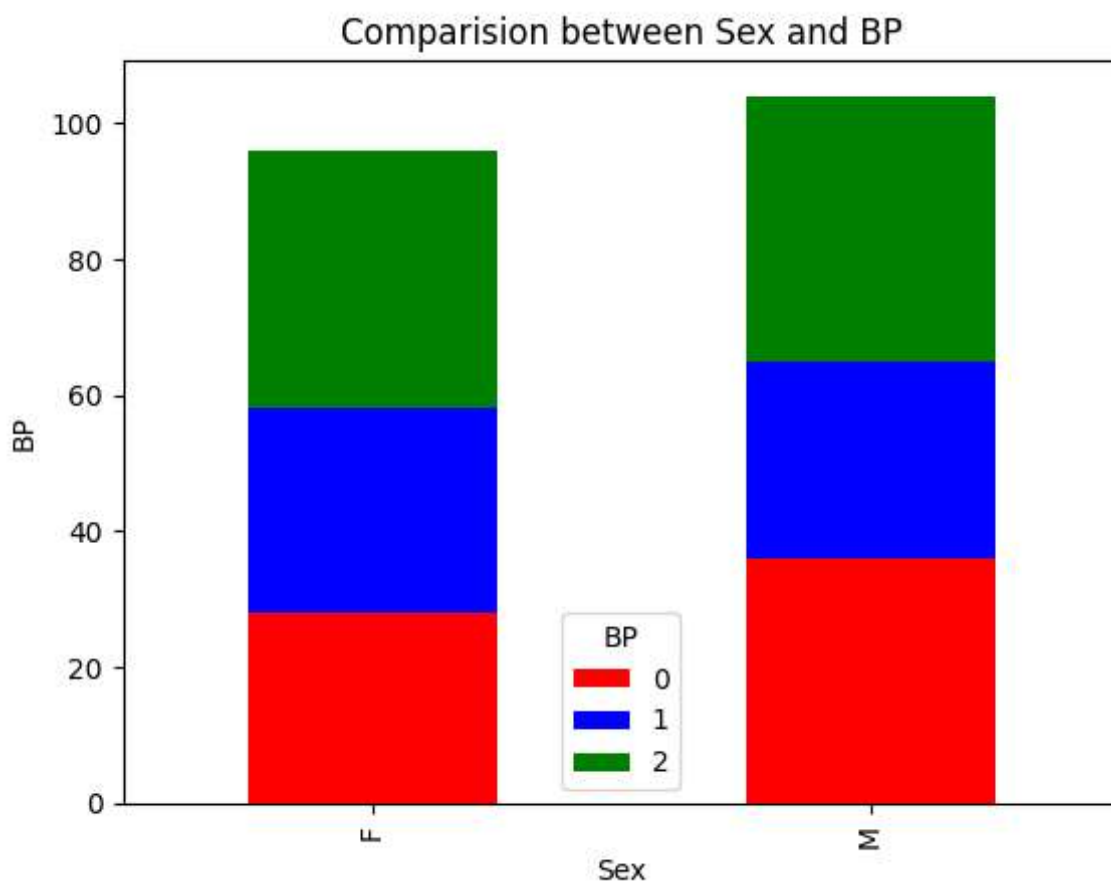
In [46]:

```
s=pd.crosstab(df['Sex'],df['BP'])  
print(s)  
s.plot(kind='bar',stacked=True,color=['red','blue','green'],grid=False)  
plt.xlabel('Sex')  
plt.ylabel('BP')  
plt.title('Comparision between Sex and BP')
```

BP	0	1	2
Sex			
F	28	30	38
M	36	29	39

Out[46]:

Text(0.5, 1.0, 'Comparision between Sex and BP')



In [47]:

```

convert={"BP":{"LOW":0,"HIGH":2,"NORMAL":1}}
df=df.replace(convert)
print(df)

```

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	2	1	25.355	5
1	47	M	0	1	13.093	3
2	47	M	0	1	10.114	3
3	28	F	1	1	7.798	4
4	61	F	0	1	18.043	5
..
195	56	F	0	1	11.567	3
196	16	M	0	1	12.006	3
197	52	M	1	1	9.894	4
198	23	M	1	0	14.020	4
199	40	F	0	0	11.349	4

[200 rows x 6 columns]

In [48]:

```

convert={"Cholesterol":{"HIGH":1,"NORMAL":0}}
df=df.replace(convert)
print(df)

```

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	2	1	25.355	5
1	47	M	0	1	13.093	3
2	47	M	0	1	10.114	3
3	28	F	1	1	7.798	4
4	61	F	0	1	18.043	5
..
195	56	F	0	1	11.567	3
196	16	M	0	1	12.006	3
197	52	M	1	1	9.894	4
198	23	M	1	0	14.020	4
199	40	F	0	0	11.349	4

[200 rows x 6 columns]

In [49]:

```

convert={"Drug":{"drugY":5,"drugC":3,"drugX":4,"drugA":1,"drugB":2}}
df=df.replace(convert)
print(df)

```

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	2	1	25.355	5
1	47	M	0	1	13.093	3
2	47	M	0	1	10.114	3
3	28	F	1	1	7.798	4
4	61	F	0	1	18.043	5
..
195	56	F	0	1	11.567	3
196	16	M	0	1	12.006	3
197	52	M	1	1	9.894	4
198	23	M	1	0	14.020	4
199	40	F	0	0	11.349	4

[200 rows x 6 columns]

In [50]:

```

x=[ 'Cholesterol', 'BP', 'Age']
y=["drugY" , "drugX" , "drugA","drugB","drugC"]
all_inputs=df[x]
all_classes=df["Drug"]

```

In [51]:

```

x_train,x_test,y_train,y_test = train_test_split(all_inputs,all_classes, test_size=0.4)
clt = DecisionTreeClassifier (random_state=0)
clt.fit(x_train,y_train)
clt.score(x_test,y_test)
print(score)

```

1.0

In []:

In []: