

In [1]:

```

1 import numpy as np
2 import pandas as pd
3 import seaborn as sns
4 from sklearn.model_selection import train_test_split
5 from sklearn.tree import DecisionTreeClassifier

```

In [2]:

```

1 df=pd.read_csv(r"C:\Users\HP\OneDrive\Documents\loan1.csv")
2 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]:

```
1 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: 448.0+ bytes

```

In [4]:



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

Out[4]:

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

In [5]:



```
1 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]:



```
1 convert = {"Home Owner":{"Yes":1,"No":0}}
2 df=df.replace(convert)
3 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]:



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



```
1 convert={"Defaulted Borrower":{"No":0,"Yes":1}}
2 df=df.replace(convert)
3 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]:



```
1 x=["Home Owner","Marital Status","Annual Income"]
2 y=["Yes","No"]
3 all_inputs=df[x]
4 all_classes=df["Defaulted Borrower"]
5
```

In [10]:



```
1 x_train,x_test,y_train,y_test = train_test_split(all_inputs,all_classes,test_size=0
```

In [11]:



```
1 clt=DecisionTreeClassifier(random_state=0)
```

In [12]:



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

Out[12]:

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

In [13]:



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

0.5

Drug Dataset

In [14]:



```
1 import numpy as np  
2 import pandas as pd  
3 import seaborn as sns  
4 from sklearn.model_selection import train_test_split  
5 from sklearn.tree import DecisionTreeClassifier
```

In [15]:

```
1 df=pd.read_csv(r"C:\Users\HP\OneDrive\Documents\drug200.csv")
2 df
```

Out[15]:

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

200 rows × 6 columns

In [16]:

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Age             200 non-null    int64
1   Sex             200 non-null    object
2   BP              200 non-null    object
3   Cholesterol      200 non-null    object
4   Na_to_K         200 non-null    float64
5   Drug            200 non-null    object
dtypes: float64(1), int64(1), object(4)
memory usage: 9.5+ KB
```

In [17]:



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

Out[17]:

```
Age          0
Sex          0
BP           0
Cholesterol  0
Na_to_K      0
Drug         0
dtype: int64
```

In [18]:



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

In [19]:



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

Out[19]:


```
Age
In [20]:
23 7
281 s=np.array(df['Age'])
492 s.sort()
393 print(s)
32 6
50 5
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
58 31 31 31 31 32 32 32 32 32 32 33 34 34 34 34 35 35 35 36 36 36 36 37 37
60 37 37 37 38 38 38 39 39 39 39 39 39 40 40 40 41 41 41 41 42 42 42 42 43
22 43 43 43 45 45 45 45 46 46 47 47 47 47 47 47 47 47 48 48 49 49 49 49
34 49 49 50 50 50 50 50 51 51 51 51 52 52 53 53 53 54 55 55 56 56 56 57
72 57 57 58 58 58 58 58 59 59 59 60 60 60 60 60 61 61 61 61 62 62 63 64
51 64 64 65 65 65 65 66 66 67 67 67 67 68 68 68 68 69 69 69 70 70 70 72 72
49 72 72 73 73 74 74 74 74]
26 4
24 4
In [21]:
67 4
681 np.mean(df['Age'])
61 4
56 4
Out[21]:
20 4
36 4
44.315 4
45 4
41 4
In [22]:
31 4
43 4
651 df['BP'].value_counts()
57 4
53 3
Out[22]:
40 3
BP 3
HIGH 3 77
LOW 3 64
NORMAL 3 59
Name: count, dtype: int64
69 3
35 3
In [23]:
18 3
641 df['Sex'].value_counts()
52 2
55 2
Out[23]:
19 2
Sex 2
M 104
F 96
Name: count, dtype: int64
48 2
54 1
17 1
33 1
63 1
30 1
21 1
25 1
Name: count, dtype: int64
```

In [24]:



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

Out[24]:

```
Cholesterol
HIGH      103
NORMAL     97
Name: count, dtype: int64
```

In [25]:



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

Out[25]:

```
Drug
drugY    91
drugX    54
drugA    23
drugC    16
drugB    16
Name: count, dtype: int64
```

In [26]:



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

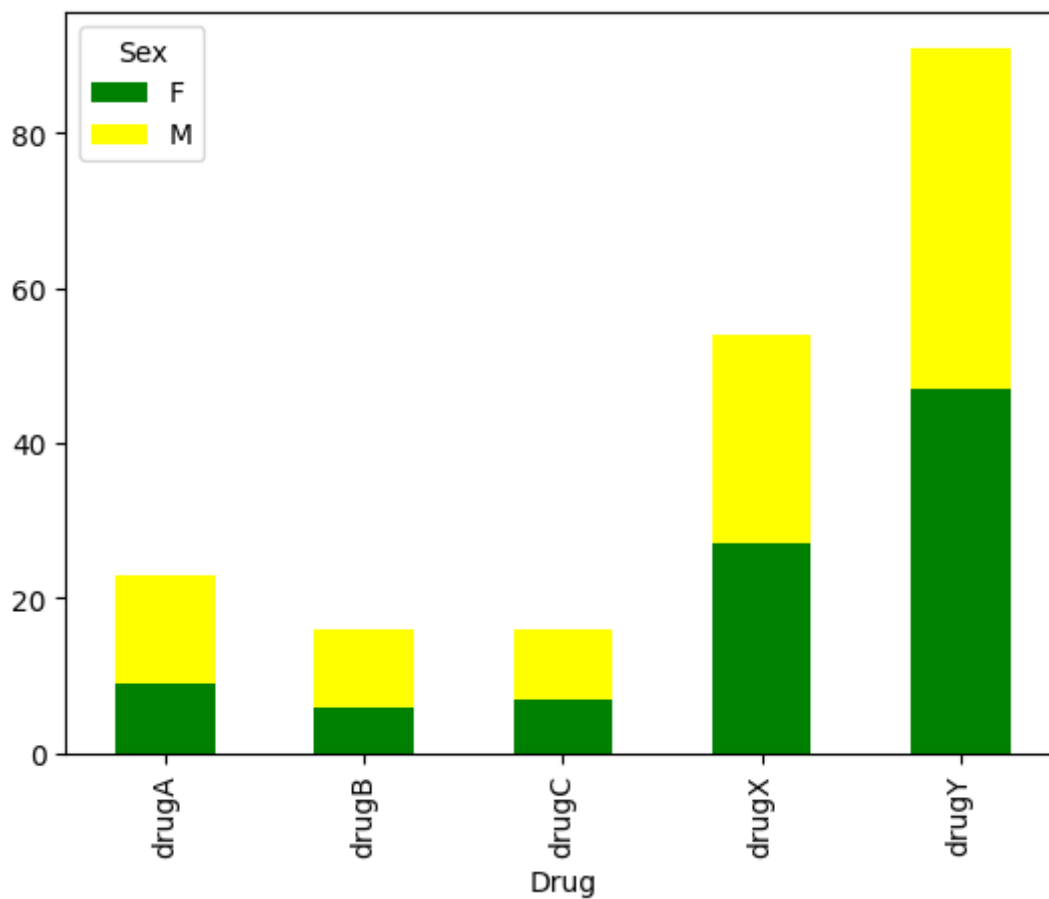
```
Sex      F      M
Drug
drugA     9     14
drugB     6     10
drugC     7      9
drugX    27     27
drugY    47     44
```

In [27]:

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

Out[27]:

<Axes: xlabel='Drug'>



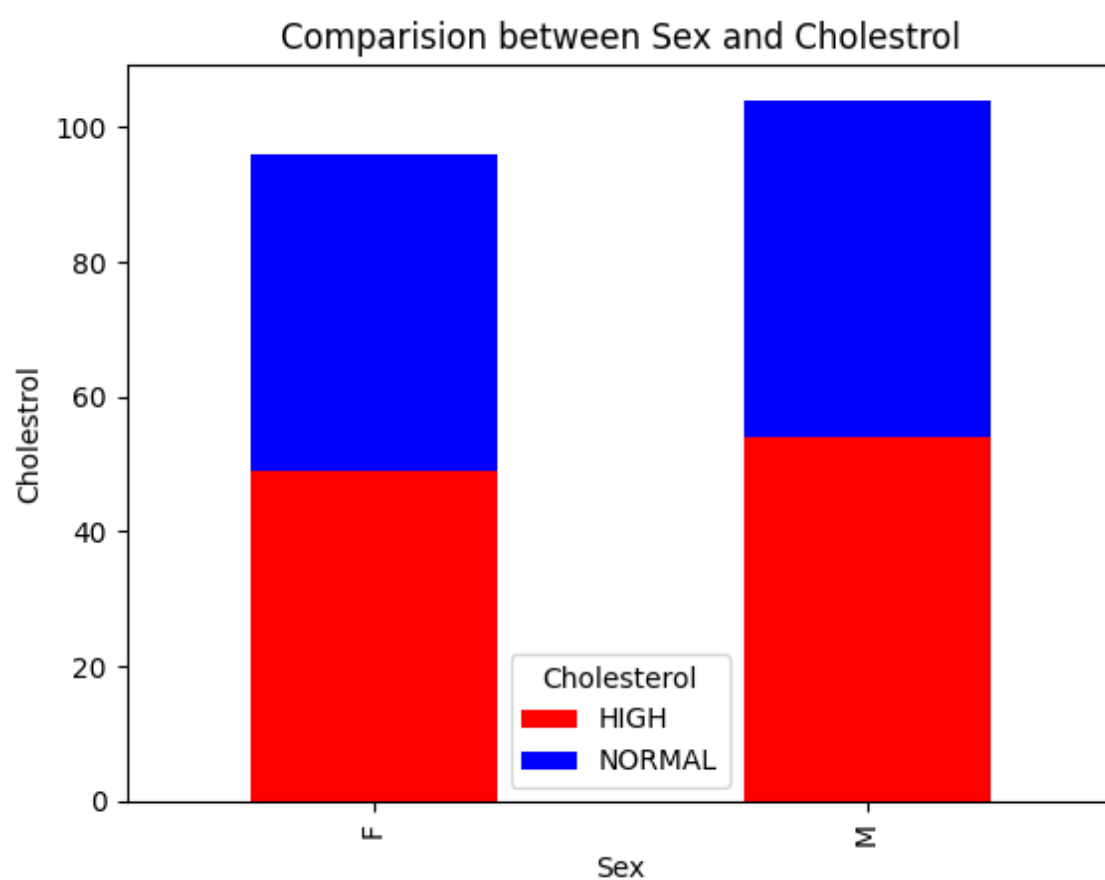
In [28]:

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

Cholesterol	HIGH	NORMAL
Sex		
F	49	47
M	54	50

Out[28]:

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



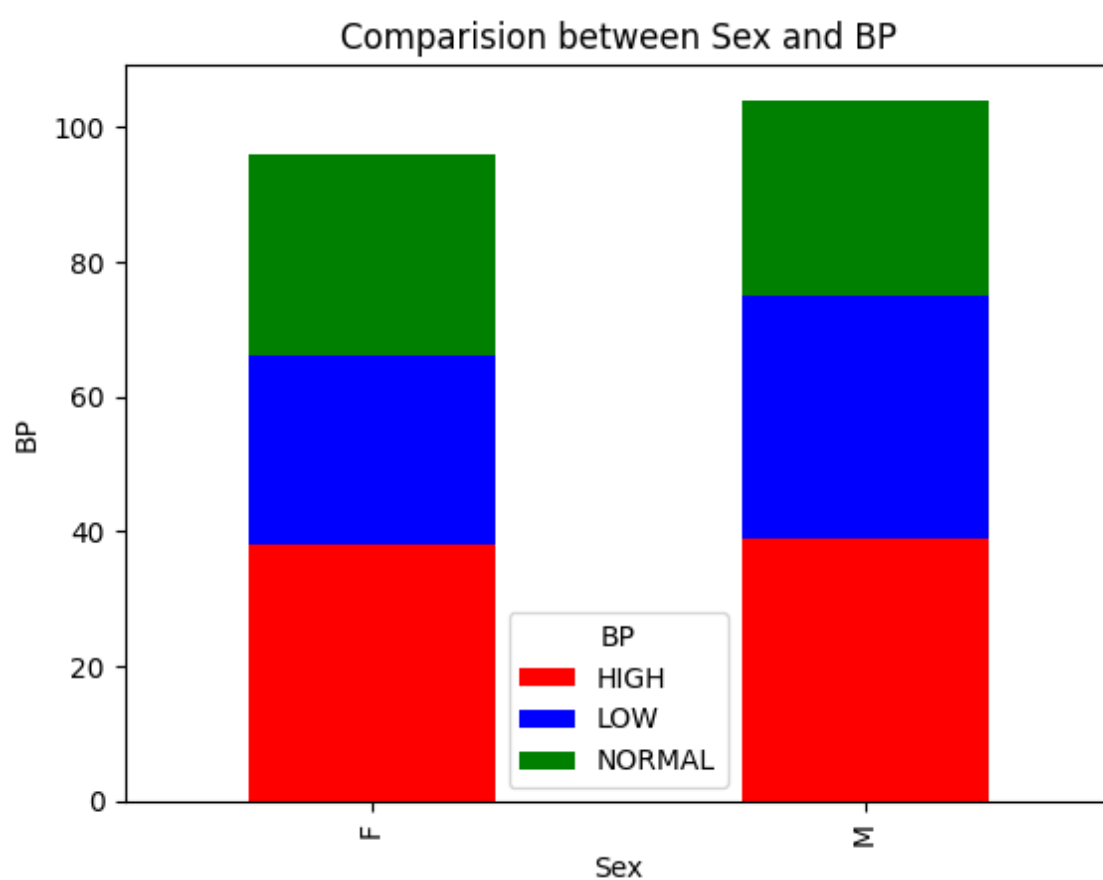
In [29]:

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

BP	HIGH	LOW	NORMAL
Sex			
F	38	28	30
M	39	36	29

Out[29]:

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



In [30]:



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

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

[200 rows x 6 columns]

In [31]:



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

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

[200 rows x 6 columns]

In [32]:



```
1 convert={"Drug":{"drugY":5,"drugC":3,"drugX":4,"drugA":1,"drugB":2}}
2 df=df.replace(convert)
3 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 [33]:



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

In [34]:



```
1 x_train,x_test,y_train,y_test = train_test_split(all_inputs,all_classes,test_size=0
2 clt=DecisionTreeClassifier(random_state=0)
3 clt.fit(x_train,y_train)
4 score=clt.score(x_test,y_test)
5 print(score)
```

0.5375

In []:



1