In [44]:

```
#Amarendra Shendkar
#MIS 111915013

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [45]:

```
df = pd.read_csv('../input/drug-classification/drug200.csv')
```

In [46]:

```
df.head()
```

Out[46]:

	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

In [47]:

```
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 [48]:

```
df.isna().sum()
```

Out[48]:

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

In [49]:

```
df = df.dropna()
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 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: 10.9+ KB
```

In [50]:

```
pd.get_dummies(df)
```

Out[50]:

	Age	Na_to_K	Sex_F	Sex_M	BP_HIGH	BP_LOW	BP_NORMAL	Cholesterol_HIGH	Cholesterol_NORMAL	Drug_DrugY	Drug_drugA	Drug_
0	23	25.355	1	0	1	0	0	1	0	1	0	
1	47	13.093	0	1	0	1	0	1	0	0	0	
2	47	10.114	0	1	0	1	0	1	0	0	0	
3	28	7.798	1	0	0	0	1	1	0	0	0	
4	61	18.043	1	0	0	1	0	1	0	1	0	
...	...	...	...	...	...	...	...	...	...	...	...	
195	56	11.567	1	0	0	1	0	1	0	0	0	
196	16	12.006	0	1	0	1	0	1	0	0	0	
197	52	9.894	0	1	0	0	1	1	0	0	0	
198	23	14.020	0	1	0	0	1	0	1	0	0	
199	40	11.349	1	0	0	1	0	0	1	0	0	

200 rows × 14 columns

In [51]:

```
X = pd.get_dummies(df.drop('Drug',axis=1),drop_first=True)
y = df['Drug']
```

In [52]:

```
X.head()
y.head()
```

Out[52]:

```
0    DrugY
1    drugC
2    drugC
3    drugX
4    DrugY
Name: Drug, dtype: object
```

In [53]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)
```

In [54]:

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
```

In [55]:

```
scaled_X_train = scaler.fit_transform(X_train)
scaled_X_test = scaler.transform(X_test)
```

In [56]:

```
from sklearn.neighbors import KNeighborsClassifier
```

In [57]:

```
knn_model = KNeighborsClassifier(n_neighbors=10)
```

In [58]:

```
knn_model.fit(scaled_X_train,y_train)
```

Out[58]:

```
KNeighborsClassifier(n_neighbors=10)
```

In [59]:

```
full_test = pd.concat([X_test,y_test],axis=1)
len(full_test)
```

Out[59]:

```
60
```

In [60]:

```
y_pred = knn_model.predict(scaled_X_test)
```

In [61]:

```
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
```

In [62]:

```
accuracy_score(y_test,y_pred)
```

Out[62]:

```
0.7666666666666667
```

In [63]:

```
confusion_matrix(y_test,y_pred)
```

Out[63]:

```
array([[21,  1,  0,  0,  4],
       [ 0,  6,  0,  0,  0],
       [ 0,  1,  5,  0,  0],
       [ 6,  0,  0,  3,  0],
       [ 2,  0,  0,  0, 11]])
```

In [64]:

```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

       DrugY       0.72      0.81      0.76        26
       drugA       0.75      1.00      0.86         6
       drugB       1.00      0.83      0.91         6
       drugC       1.00      0.33      0.50         9
       drugX       0.73      0.85      0.79        13

    accuracy                           0.77        60
   macro avg       0.84      0.76      0.76        60
weighted avg       0.80      0.77      0.75        60
```