

IMPORTING LIBRARIES

In [1]:

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

IMPORTING DATASET

In [2]:

```
ds=pd.read_csv('heart.csv')
x=ds.iloc[:, :-1].values
y=ds.iloc[:, 13].values
```

STANDARDIZING 'X'

In [3]:

```
from sklearn.preprocessing import StandardScaler
sc= StandardScaler()
x=sc.fit_transform(x)
```

SPLITTING THE DATASET INTO TRAIN AND TEST DATA

In [4]:

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(x,y,test_size=0.15,random_state=0)
```

APPLYING 'PCA'

In [5]:

```
from sklearn.decomposition import PCA
pca=PCA(n_components=2)
X_train=pca.fit_transform(X_train)
X_test=pca.transform(X_test)
explained_variance=pca.explained_variance_ratio_
print(explained_variance)
```

```
[0.20948567 0.12122518]
```

TRAINING THE K-NN MODEL ON TRAINING SET

In [6]:

```
from sklearn.neighbors import KNeighborsClassifier
clas=KNeighborsClassifier(n_neighbors =6,metric='minkowski',p=2)
clas.fit(X_train,y_train)
```

Out[6]:

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                    metric_params=None, n_jobs=None, n_neighbors=6, p=2,
                    weights='uniform')
```

PREDICTING THE TEST RESULTS

In [7]:

```
y_pred=clas.predict(X_test)
#print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1))
```

CONFUSION MATRIX AND ACCURACY

In [8]:

```
from sklearn.metrics import confusion_matrix,accuracy_score
cm = confusion_matrix(y_test,y_pred)
print(cm)
accuracy_score(y_test,y_pred)
```

```
[[22  1]
 [ 3 20]]
```

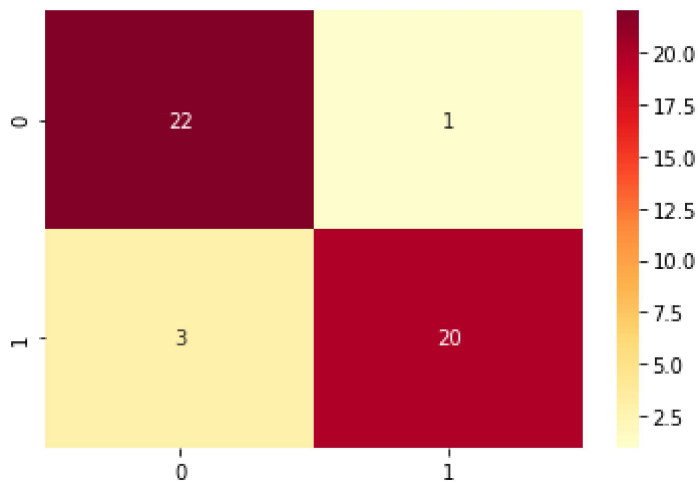
Out[8]:

```
0.9130434782608695
```

PLOT OF CONFUSION MATRIX

In [9]:

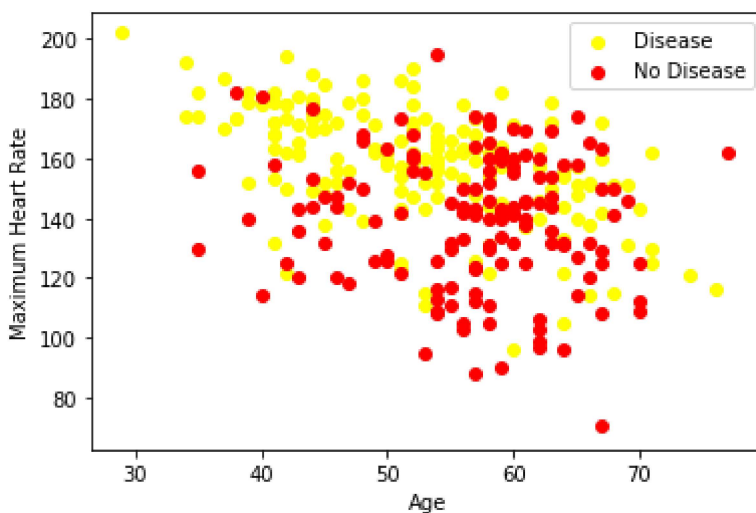
```
import seaborn as sns
sns.heatmap(cm,annot=True,cmap="YlOrRd")
plt.show()
```



PLOT BETWEEN AGE AND MAX. HEART RATE

In [10]:

```
plt.scatter(x=ds.age[ds.target==1], y=ds.thalach[(ds.target==1)], c="yellow")
plt.scatter(x=ds.age[ds.target==0], y=ds.thalach[(ds.target==0)], c = 'red')
plt.legend(["Disease", "No Disease"])
plt.xlabel("Age")
plt.ylabel("Maximum Heart Rate")
plt.show()
```



COUNT OF MALE AND FEMALE

In [11]:

```
pd.crosstab(ds.sex,ds.target).plot(kind="bar",figsize=(10,5),color=['#1CA53B','#EE0000'])  
plt.title('Heart Disease Frequency for Sex')  
plt.xlabel('Sex (0 = Female,1 = Male)')  
plt.xticks(rotation=0 )  
plt.legend(["No Disease", " Disease "])  
plt.ylabel("Frequency")  
plt.show()
```

