

Import the Libraries

In [1]:

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

Import the Dataset

In [2]:

```
dataset=pd.read_csv('heart.csv')
```

Extracting x and y

In [3]:

```
x=dataset.iloc[:, :-1].values
y=dataset.iloc[:, -1].values
```

Standardizing the x

In [4]:

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

Splitting the dataset into train and test data

In [5]:

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

Applying the PCA

In [6]:

```
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)
```

```
[0.20879927 0.1211517 ]
```

Training the svm model on training set

In [7]:

```
from sklearn.svm import SVC
classifier=SVC(kernel='linear',random_state=0)
classifier.fit(X_train,y_train)
```

Out [7]:

```
Out[7]:
```

```
SVC(kernel='linear', random_state=0)
```

Predicting the Test Results

```
In [8]:
```

```
from sklearn import metrics
y_pred=classifier.predict(X_test)
print(y_pred)
```

```
[0 1 1 0 0 1 0 0 0 0 1 1 0 1 1 0 0 1 0 0 0 1 0 0 1 1 1 0 0 1 1 1 0 0
 1 0 0 1 1 1 0 0 1 1 1 0 1 1 0 1 1 1 0 1 1 1 1]
```

Calculating the Accuracy

```
In [9]:
```

```
accuracy=metrics.accuracy_score(y_test,y_pred)
print('Accuracy: {:.2f}'.format(accuracy))
```

```
Accuracy: 0.84
```

Creating the Confusion Matrix

```
In [10]:
```

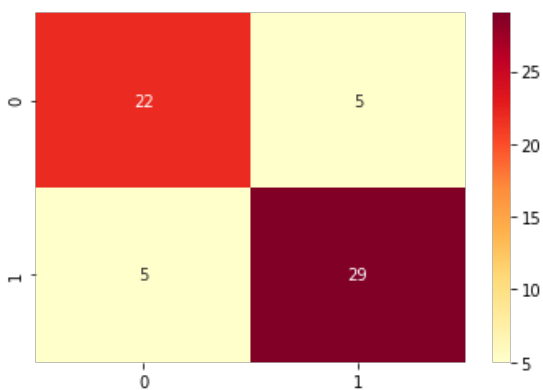
```
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,y_pred)
print(cm)
```

```
[[22  5]
 [ 5 29]]
```

Plotting the Confusion Matrix

```
In [11]:
```

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

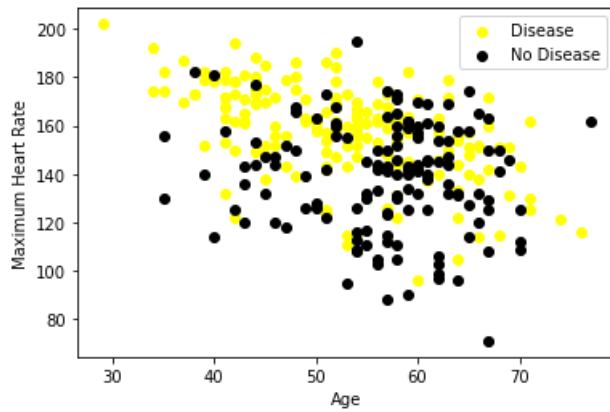


Scatter plot between Age and Max Heart Beat

```
In [12]:
```

```
plt.scatter(x=dataset.age[dataset.target==1], y=dataset.thalach[(dataset.target==1)], c="yellow")
```

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

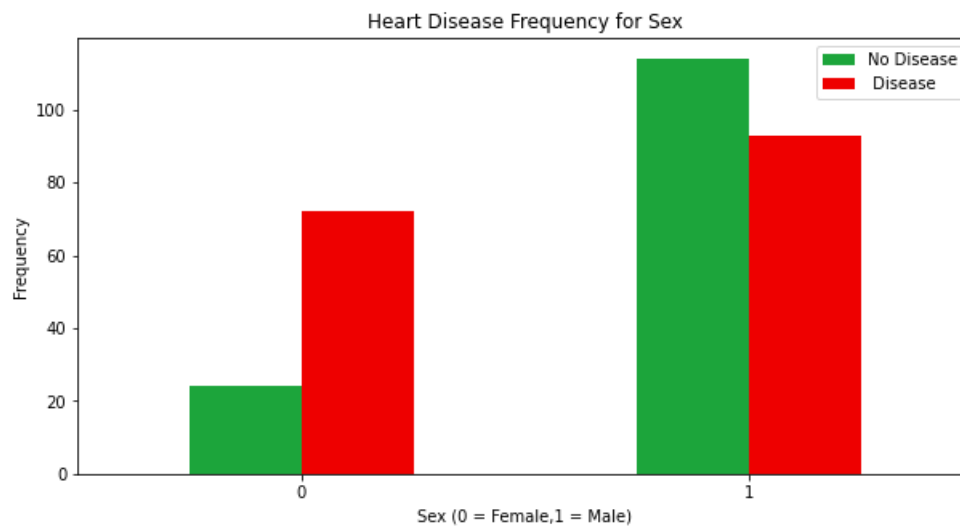


Bar Plot for count of male and female

In [16]:

```
pd.crosstab(dataset.sex,dataset.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()
```



In []: