Heart Disease Prediction

I am using Heart disease dataset downloaded from kaggle website. This dataset has target value which tells whether disease is present or not . We we will be using this dataset by splitting it into train and test model

In [39]:
```
#Feature is dataset (Metadata information)
#Age: The person's age in years

#Sex: The person's sex (1 = male, 0 = female)

#Chest pain type: The chest pain experienced (Value 1: typical angina, Value 2: atypical

#Bp: The person's resting blood pressure (mm Hg on admission to the hospital)

#chol: The person's cholesterol measurement in mg/dl

#FBS over 120: The person's fasting blood sugar (> 120 mg/dl, 1 = true; 0 = false)

#EKG Results: Resting electrocardiographic measurement (0 = normal, 1 = having ST-T wave

#Max HR: The person's maximum heart rate achieved

#Exercise induced angina: Exercise induced angina (1 = yes; 0 = no)

#ST depression: ST depression induced by exercise relative to rest ('ST' relates to posi

#Slope of ST: the slope of the peak exercise ST segment (Value 1: upsloping, Value 2: fl

#The number of vessel: The number of major vessels (0-3)

#Thalassemia: A blood disorder called thalassemia (3 = normal; 6 = fixed defect; 7 = rev

#Heart Disease: Heart disease (0 = no, 1 = yes)
```

1. Importing all necessary libraries :

In [40]:
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
from sklearn import metrics
from sklearn.ensemble import AdaBoostClassifier
from sklearn.metrics import ConfusionMatrixDisplay
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

2. Importing Dataset :

In [41]:
```
df=pd.read_csv(r'C:\Users\kmishra1\Desktop\kritika\KaggleDataset\Heart_Disease_Pred.csv'
df1=df
df1 = pd.get_dummies(df1, columns=['Heart Disease'], drop_first=True)
df1.rename(columns={'Heart Disease_Presence':'Heart Disease'}, inplace=True)
df1.rename(columns={'Thallium':'Thalesemia'},inplace=True)
```

3. Exploratory Data Analysis

In [42]:
```
#To check number column in dataframe df1

df1.head()
```

Out[42]:

	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels fluro	Thalesemia	H Dis
0	70	1	4	130	322	0	2	109	0	2.4	2	3	3	
1	67	0	3	115	564	0	2	160	0	1.6	2	0	7	
2	57	1	2	124	261	0	0	141	0	0.3	1	0	7	
3	64	1	4	128	263	0	0	105	1	0.2	2	1	7	
4	74	0	2	120	269	0	2	121	1	0.2	1	1	3	

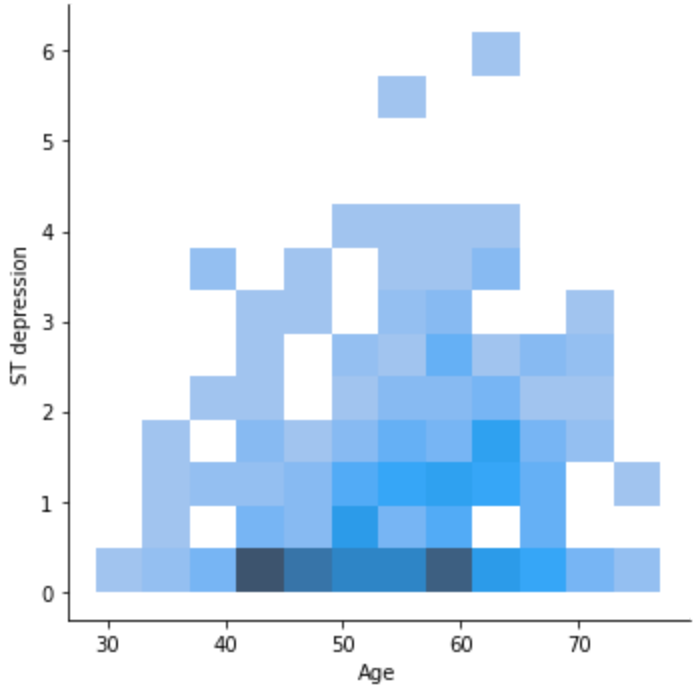
In [43]:
```
# To Check presence of NA or NULL values if any in data
# To do so i will be using isna() inbuilt function in python and sum() function to get t

df1.isna().sum()
```

Out[43]:
```
Age                       0
Sex                       0
Chest pain type           0
BP                        0
Cholesterol               0
FBS over 120              0
EKG results               0
Max HR                    0
Exercise angina           0
ST depression             0
Slope of ST               0
Number of vessels fluro   0
Thalesemia                0
Heart Disease             0
dtype: int64
```

Plotting Displot to check the distribution of ST depression across all the age groups

In [44]:
```
plt.figure(figsize=(10,9))
sns.displot(df1,x='Age',y='ST depression')
```

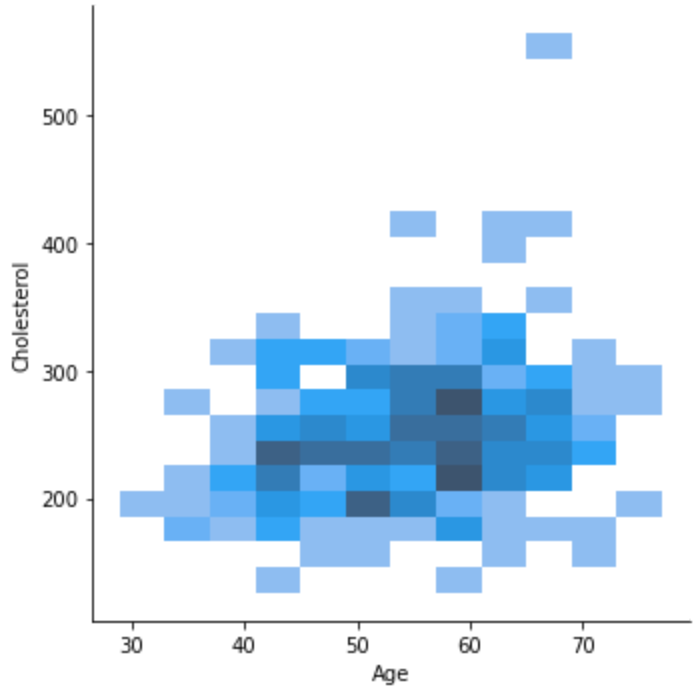
Out[44]:
```
<seaborn.axisgrid.FacetGrid at 0x2a822428e80>

<Figure size 720x648 with 0 Axes>
```



Plotting Displot to check the distribution of Cholestrol across all the age groups

In [45]:
```
plt.figure(figsize=(5,3))
sns.displot(df1,x='Age',y='Cholesterol')
```

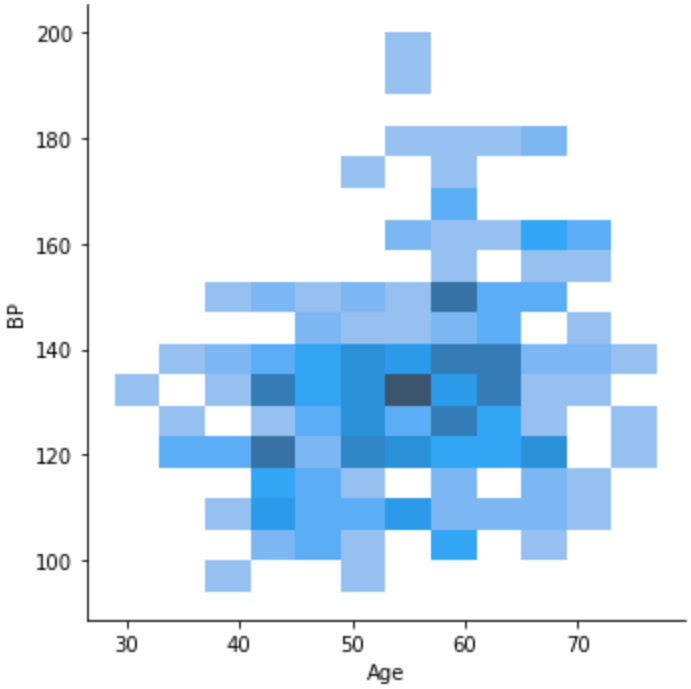
Out[45]:
```
<seaborn.axisgrid.FacetGrid at 0x2a822714a90>

<Figure size 360x216 with 0 Axes>
```



Plotting Displot to check the distribution of BP across all the age groups

In [46]:
```
plt.figure(figsize=(5,5))
sns.displot(df1,x='Age',y='BP')
```

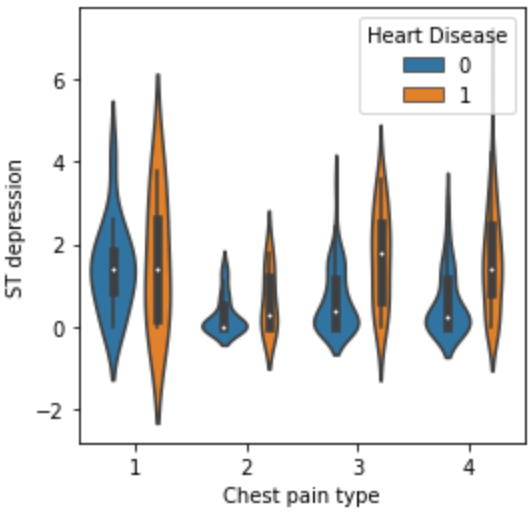
Out[46]:
```
<seaborn.axisgrid.FacetGrid at 0x2a81f8ac760>

<Figure size 360x360 with 0 Axes>
```



I will be using violin plot to understand density distribution of data with respect to heart disease and chest pain type vs ST depression

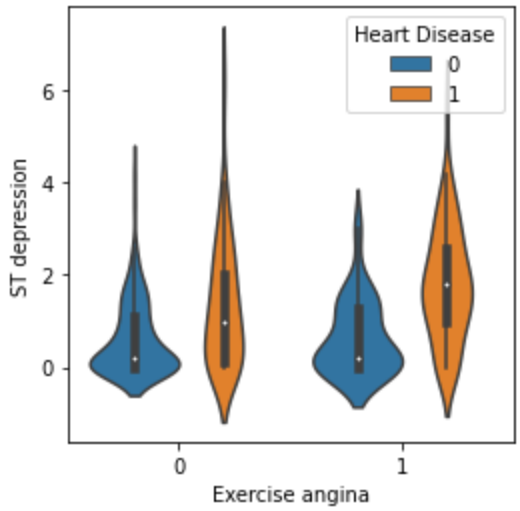
In [47]:
```
plt.figure(figsize=(4,4))
sns.violinplot(data=df1,x='Chest pain type',y='ST depression',hue='Heart Disease')
```

Out[47]:
```
<AxesSubplot:xlabel='Chest pain type', ylabel='ST depression'>
```



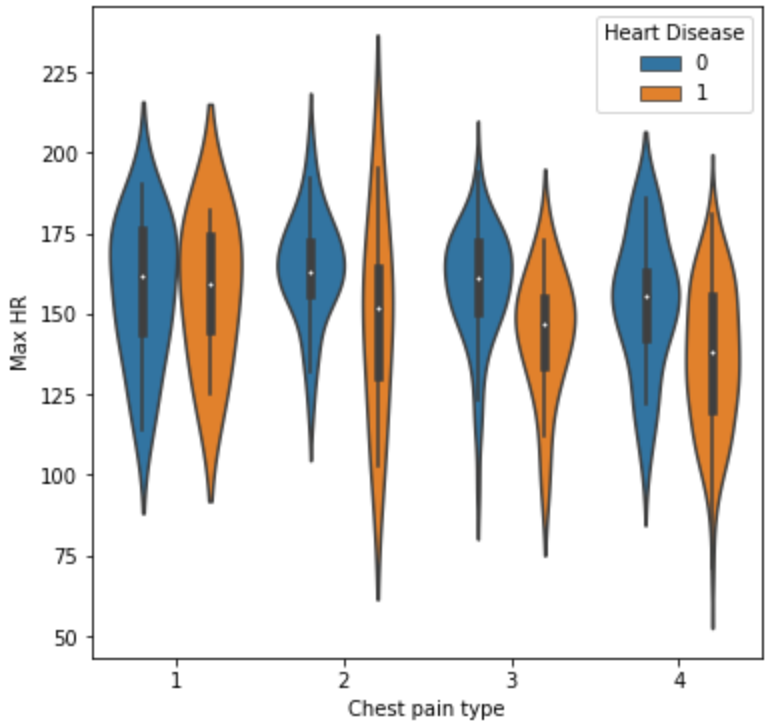
In [48]:
```
plt.figure(figsize=(4,4))
sns.violinplot(data=df1,x='Exercise angina',y='ST depression',hue='Heart Disease')
```

Out[48]:
```
<AxesSubplot:xlabel='Exercise angina', ylabel='ST depression'>
```



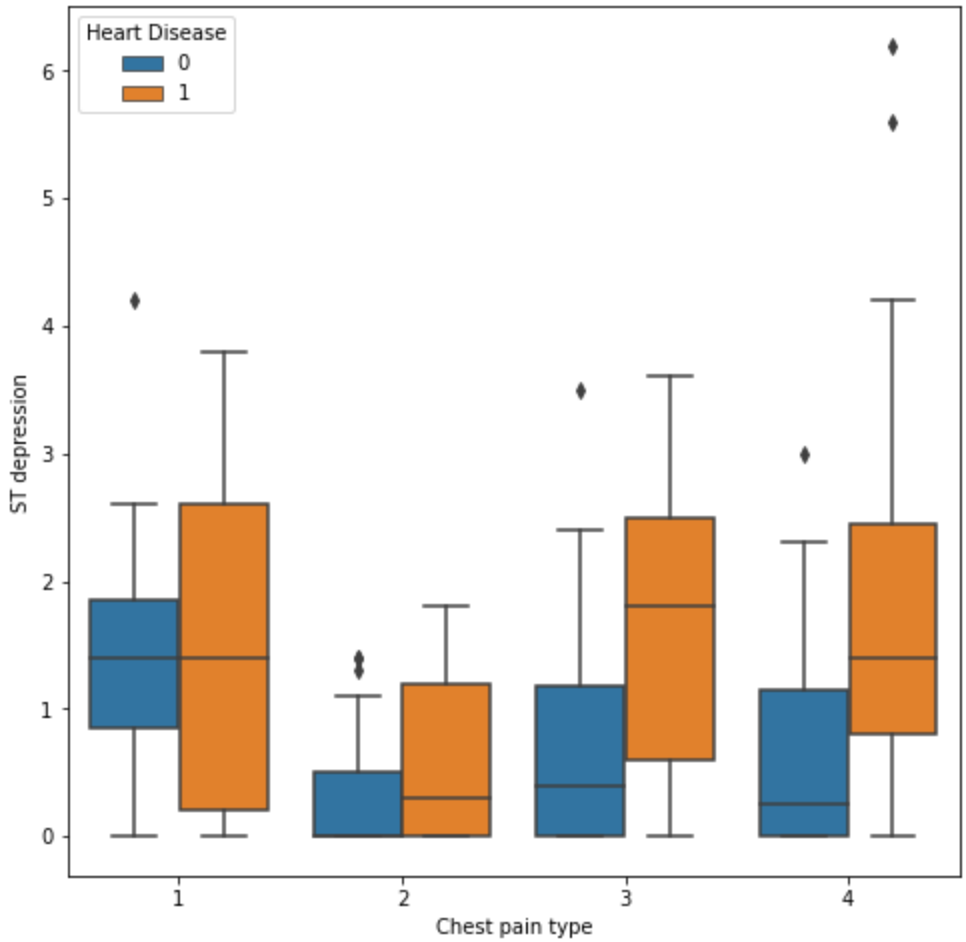
In [49]:
```
plt.figure(figsize=(6,6))
sns.violinplot(data=df1,x='Chest pain type',y='Max HR',hue='Heart Disease')
```

Out[49]:
```
<AxesSubplot:xlabel='Chest pain type', ylabel='Max HR'>
```



In [50]:
```
plt.figure(figsize=(8,8))
sns.boxplot(data=df1,x='Chest pain type',y='ST depression',hue='Heart Disease',dodge=Tru
```

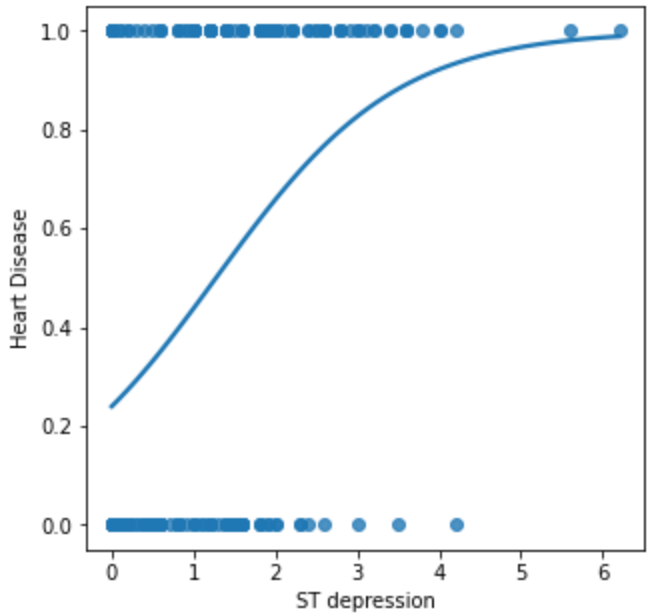
Out[50]:
```
<AxesSubplot:xlabel='Chest pain type', ylabel='ST depression'>
```



Plotting logictic regression relations graphs between Predictor and Variable

In [51]:
```
plt.figure(figsize=(5,5))
x=df1['ST depression']
y=df1['Heart Disease']

sns.regplot(x=x, y=y, data=df1, logistic=True, ci=None)
```

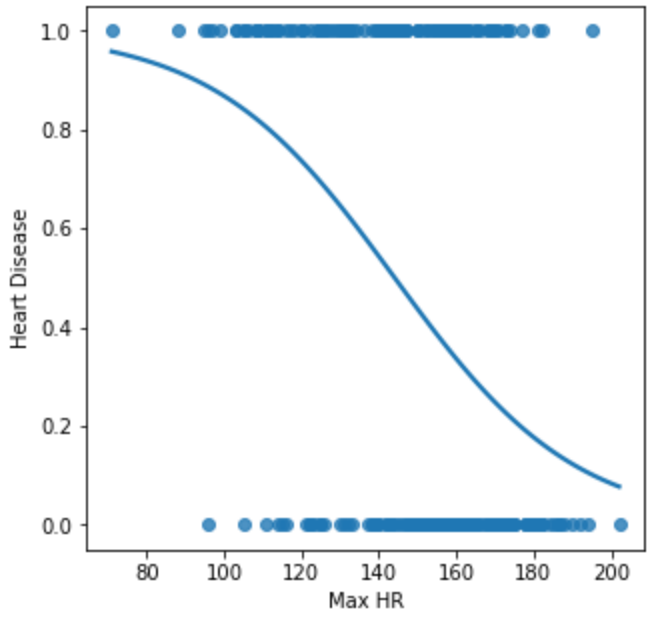
Out[51]:
```
<AxesSubplot:xlabel='ST depression', ylabel='Heart Disease'>
```



In [52]:
```
plt.figure(figsize=(5,5))
x=df1['Max HR']
y=df1['Heart Disease']
```

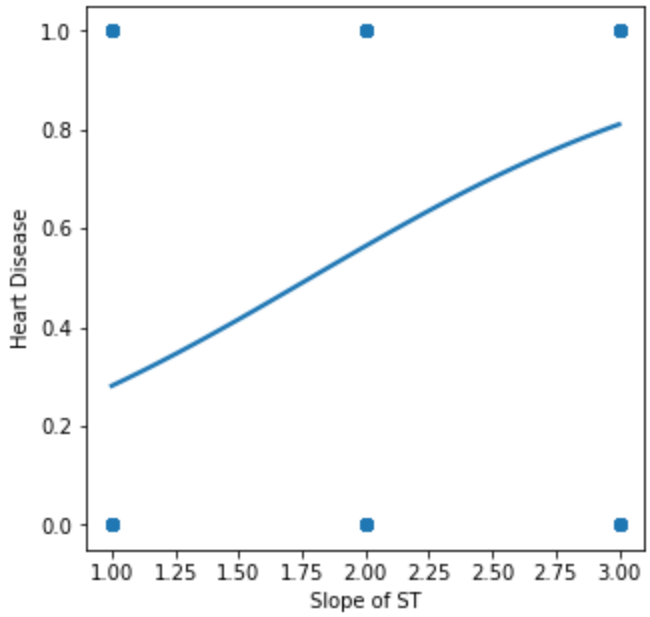
```
sns.regplot(x=x, y=y, data=df1, logistic=True, ci=None)
```

Out[52]:
```
<AxesSubplot:xlabel='Max HR', ylabel='Heart Disease'>
```



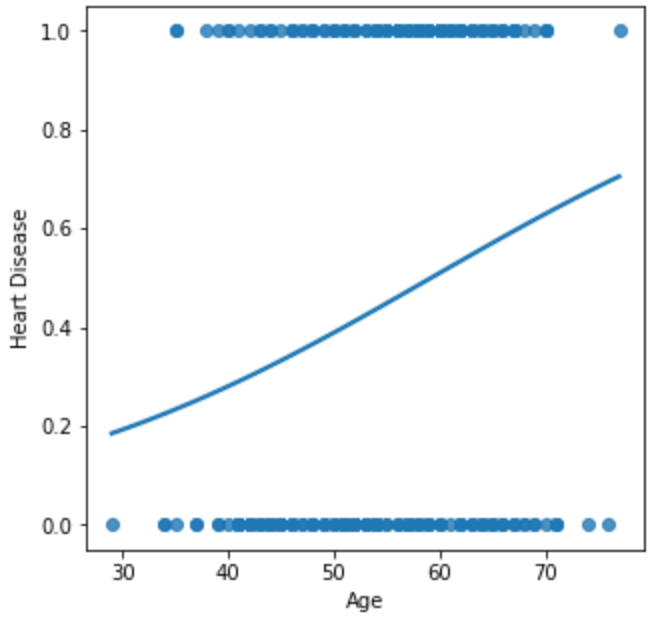
In [53]:
```
plt.figure(figsize=(5,5))
x=df1['Slope of ST']
y=df1['Heart Disease']

sns.regplot(x=x, y=y, data=df1, logistic=True, ci=None)
```

Out[53]:
```
<AxesSubplot:xlabel='Slope of ST', ylabel='Heart Disease'>
```



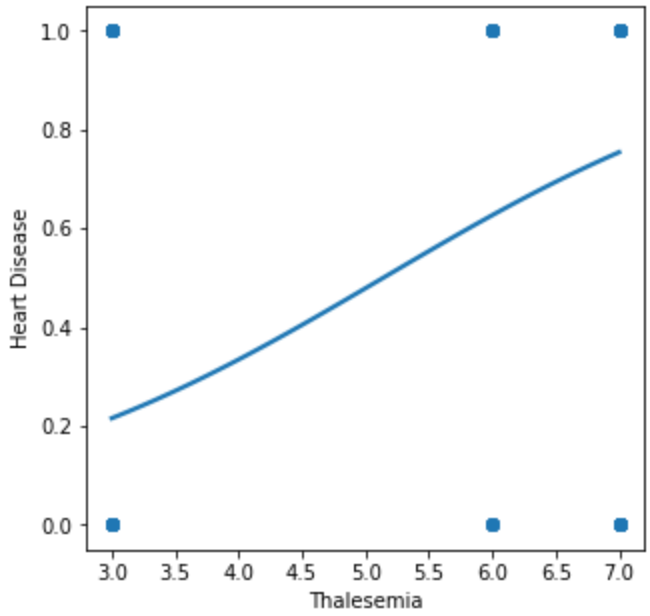
In [54]:
```
plt.figure(figsize=(5,5))
x=df1['Age']
y=df1['Heart Disease']

sns.regplot(x=x, y=y, data=df1, logistic=True, ci=None)
```

Out[54]:
```
<AxesSubplot:xlabel='Age', ylabel='Heart Disease'>
```



In [55]:
```
plt.figure(figsize=(5,5))
x=df1['Thalesemia']
y=df1['Heart Disease']

sns.regplot(x=x, y=y, data=df1, logistic=True, ci=None)
```

Out[55]:
```
<AxesSubplot:xlabel='Thalesemia', ylabel='Heart Disease'>
```



As we can see the heart rate has negetive correlation with Max HR (Max Heart Rate acheived) . However there is postive corelation between ST Depression , Chest Pain type , Slope of ST & Thalesemia and there is slight corelation with Age.

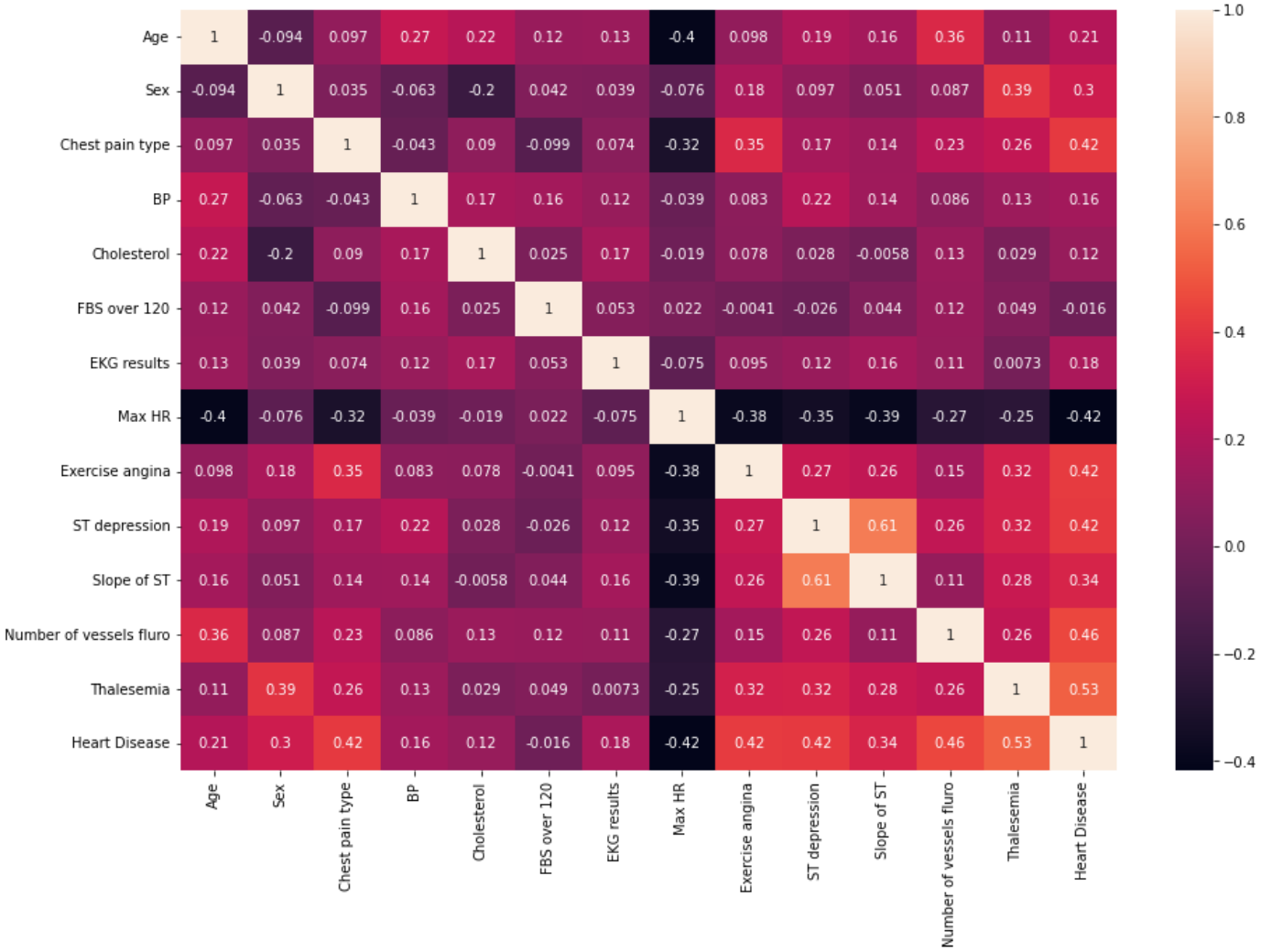
Here ST Depression believed as a common electrocardiographic sign of myocardial ischemia during exercise testing. Ischemia is generally defined as oxygen deprivation due to reduced perfusion

Exercise-induced ST-elevation is extremely uncommon especially in patients without prior myocardial infarction.

In [56]:
```
#To look for pairwise corelation among all variable we will be using heatmaap from sea b
```

```
plt.figure(figsize=(15,10))
sns.heatmap(df1.corr(),annot=True)
```

Out[56]: <AxesSubplot:>



4. Prediction of Heart diease present or not by using ADABoast Classifier

We will be using ADABoast classifier for classification of dieaseas present or not present(present : 1 and normal : 0)

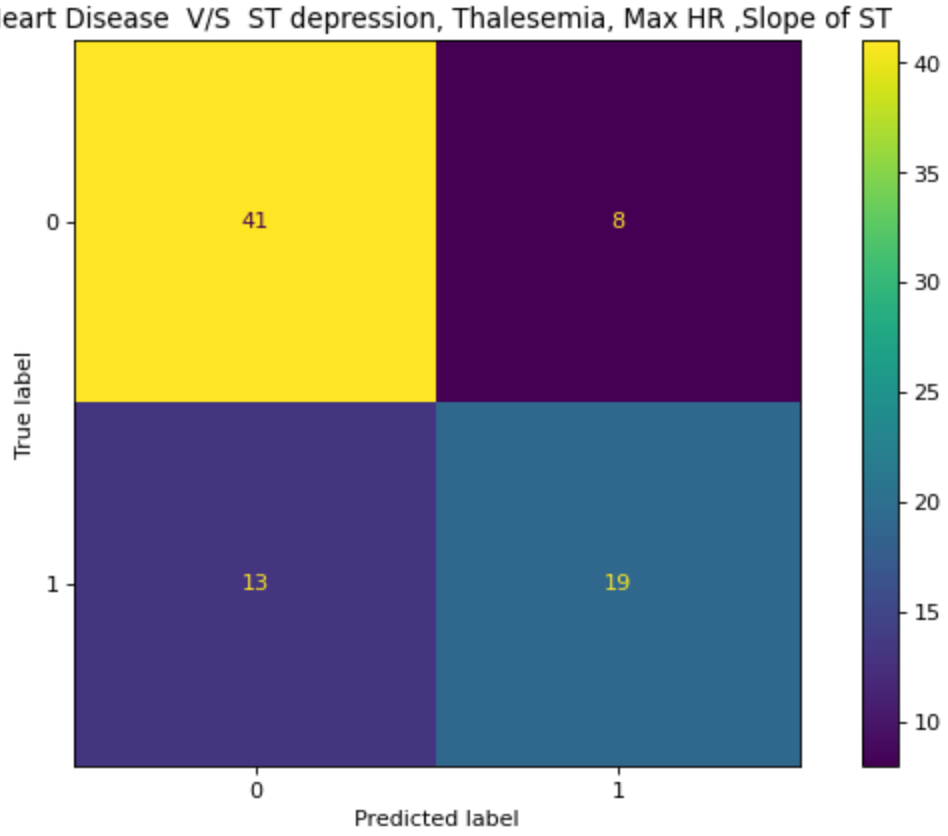
It combines multiple weak classifiers to increase the accuracy of classifiers

Here we will be using test size of 0.3 That is 70 % will be training dataset and 30% will be test dataset

In [65]:
```
# Lets check the accuracy with ADA boast classifier and classify the data


X1=df1[['ST depression','Thalesemia','Max HR','Slope of ST']]
y1=df1['Heart Disease']


#With sklearn.model_selection.train_test_split I will be creating 4 portions of data whi
X_train,X_test,y_train,y_test = train_test_split(X1,y1,test_size=0.3,random_state=42)

#Create adaboost classifer object
abc = AdaBoostClassifier( random_state=1)
model = abc.fit(X_train,y_train)
```

```
# predict the response to test data

prediction = model.predict(X_test)

print("AdaBoost Classifier Model Accuracy:", accuracy_score(y_test, prediction))



#Creating Confusion matrix
confusion_matrix1=metrics.confusion_matrix(y_test, prediction)
confusion_matrix1


print('The accuracy Score of Heart Disease  %s',accuracy_score(y_test, prediction).round

from sklearn.metrics import ConfusionMatrixDisplay

fig, ax = plt.subplots(figsize=(10,6), dpi=80)
display = ConfusionMatrixDisplay(confusion_matrix1)
ax.set(title='Heart Disease  V/S  ST depression, Thalesemia, Max HR ,Slope of ST')
display.plot(ax=ax);
```

```
AdaBoost Classifier Model Accuracy: 0.7407407407407407
The accuracy Score of Heart Disease  %s 0.74
```



Here we see the predictor has predicted 74% acurracy rate with ADABoost clasifier algorithm which is quite OK .

Also the truth table is showing 19 predictor to be having presence of heart disease where 41 showing as no heart diease with ST depression, Thalesemia, Max HR ,Slope of ST predictor variable