Mini Project by:

Saneet Saluja, Omkar Londhe, Rusheil Baath, Swarup Satav, Jatin Garad

Panel 2 CSE AI-DS Batch: B1

```
import pandas as pd
import numpy as np
```

```
#importing datasets
df=pd.read_csv("/content/heart-statlog_csv.csv")
df
```

	age	sex	chest	resting_blood_pressure	serum_cholestoral	fasting_blood_sugar	re
0	70	1	4	130	322	0	
1	67	0	3	115	564	0	
2	57	1	2	124	261	0	
3	64	1	4	128	263	0	
4	74	0	2	120	269	0	
...	...	...	...	...	...	...	
265	52	1	3	172	199	1	
266	44	1	2	120	263	0	
267	56	0	2	140	294	0	
268	57	1	4	140	192	0	
269	67	1	4	160	286	0	

270 rows × 14 columns

```
#display first 5 records
df.head()
```

	age	sex	chest	resting_blood_pressure	serum_cholestoral	fasting_blood_sugar	resting_electrocard
0	70	1	4	130	322	0	
1	67	0	3	115	564	0	
2	57	1	2	124	261	0	
3	64	1	4	128	263	0	
4	74	0	2	120	269	0	

```
#display last 5 records
df.tail()
```

	age	sex	chest	resting_blood_pressure	serum_cholestoral	fasting_blood_sugar	resting_electroca
265	52	1	3	172	199	1	
266	44	1	2	120	263	0	
267	56	0	2	140	294	0	
268	57	1	4	140	192	0	
269	67	1	4	160	286	0	

```
#show all attributes with data types
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 270 entries, 0 to 269
Data columns (total 14 columns):
 #   Column                                Non-Null Count  Dtype  
---  ------                                --------------  -----  
 0   age                                   270 non-null    int64  
 1   sex                                   270 non-null    int64  
 2   chest                                 270 non-null    int64  
 3   resting_blood_pressure                270 non-null    int64  
 4   serum_cholestoral                     270 non-null    int64  
 5   fasting_blood_sugar                   270 non-null    int64  
 6   resting_electrocardiographic_results  270 non-null    int64  
 7   maximum_heart_rate_achieved           270 non-null    int64  
 8   exercise_induced_angina               270 non-null    int64  
 9   oldpeak                               270 non-null    float64
 10  slope                                 270 non-null    int64  
 11  number_of_major_vessels               270 non-null    int64  
 12  thal                                  270 non-null    int64  
 13  class                                 270 non-null    object 
dtypes: float64(1), int64(12), object(1)
memory usage: 29.7+ KB
```

```
#show statistical info
df.describe()
```

	age	sex	chest	resting_blood_pressure	serum_cholestoral	fasting_blood_suga
count	270.000000	270.000000	270.000000	270.000000	270.000000	270.00000
mean	54.433333	0.677778	3.174074	131.344444	249.659259	0.14814
std	9.109067	0.468195	0.950090	17.861608	51.686237	0.35590
min	29.000000	0.000000	1.000000	94.000000	126.000000	0.00000
25%	48.000000	0.000000	3.000000	120.000000	213.000000	0.00000
50%	55.000000	1.000000	3.000000	130.000000	245.000000	0.00000
75%	61.000000	1.000000	4.000000	140.000000	280.000000	0.00000
max	77.000000	1.000000	4.000000	200.000000	564.000000	1.00000

```
#display data types of attributes
df.dtypes
```

```
age                                     int64
sex                                     int64
chest                                   int64
resting_blood_pressure                  int64
serum_cholestoral                       int64
fasting_blood_sugar                     int64
resting_electrocardiographic_results    int64
maximum_heart_rate_achieved             int64
exercise_induced_angina                 int64
oldpeak                               float64
slope                                   int64
number_of_major_vessels                 int64
thal                                    int64
class                                  object
dtype: object
```

```
#to get the column labels of dataframe
df.columns
```

```
Index(['age', 'sex', 'chest', 'resting_blood_pressure', 'serum_cholestoral',
       'fasting_blood_sugar', 'resting_electrocardiographic_results',
       'maximum_heart_rate_achieved', 'exercise_induced_angina', 'oldpeak',
       'slope', 'number_of_major_vessels', 'thal', 'class'],
      dtype='object')
```

```
#get the index (row labels) of dataframe
df.index
```

```
RangeIndex(start=0, stop=270, step=1)
```

```
#get the total number of elements from the data frame
df.size
```

```
3780
```

```
#get the dimensionality of data frame
df.shape
```

```
(270, 14)
```

```
#slice the result for first 5 rows
df[0:5]
```

	age	sex	chest	resting_blood_pressure	serum_cholestoral	fasting_blood_sugar	resting_electrocard
0	70	1	4	130	322	0	
1	67	0	3	115	564	0	
2	57	1	2	124	261	0	
3	64	1	4	128	263	0	
4	74	0	2	120	269	0	

```
#the number of axes/array dimensions
df.ndim
```

```
2
```

```
#checking for missing values
df.isnull()
```

	age	sex	chest	resting_blood_pressure	serum_cholestoral	fasting_blood_sugar	resting_electr
0	False	False	False	False	False	False	
1	False	False	False	False	False	False	
2	False	False	False	False	False	False	
3	False	False	False	False	False	False	
4	False	False	False	False	False	False	
...	...	...	...	...	...	...	
265	False	False	False	False	False	False	
266	False	False	False	False	False	False	
267	False	False	False	False	False	False	
268	False	False	False	False	False	False	
269	False	False	False	False	False	False	

270 rows × 14 columns

```
#The number of NULL values in each column
df.isnull().sum()
```

```
age                                 0
sex                                 0
chest                               0
resting_blood_pressure              0
serum_cholestoral                   0
fasting_blood_sugar                 0
resting_electrocardiographic_results    0
maximum_heart_rate_achieved         0
```

```
exercise_induced_angina               0
oldpeak                               0
slope                                 0
number_of_major_vessels               0
thal                                  0
class                                 0
dtype: int64
```

```
df.shape
```

```
(270, 14)
```

```
df.duplicated(subset=None, keep="first").value_counts()
```

```
False    270
dtype: int64
```

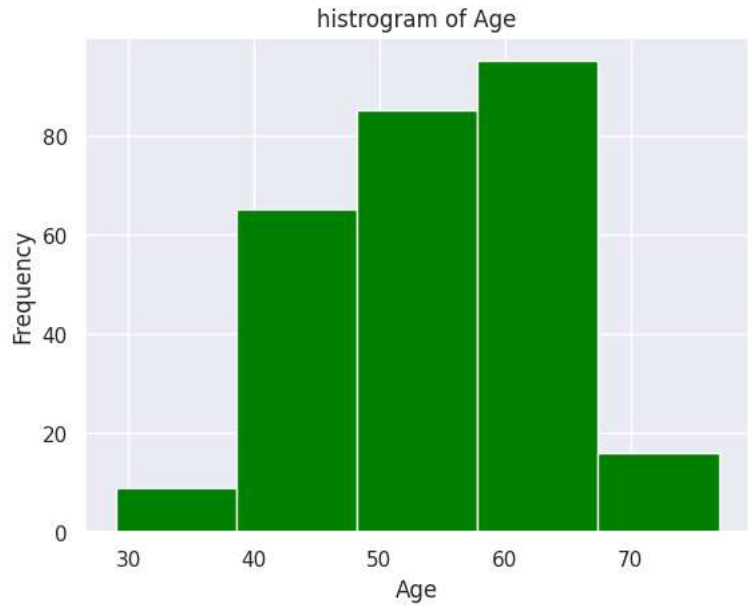
```
df.columns
```

```
Index(['age', 'sex', 'chest', 'resting_blood_pressure', 'serum_cholestoral',
       'fasting_blood_sugar', 'resting_electrocardiographic_results',
       'maximum_heart_rate_achieved', 'exercise_induced_angina', 'oldpeak',
       'slope', 'number_of_major_vessels', 'thal', 'class'],
      dtype='object')
```

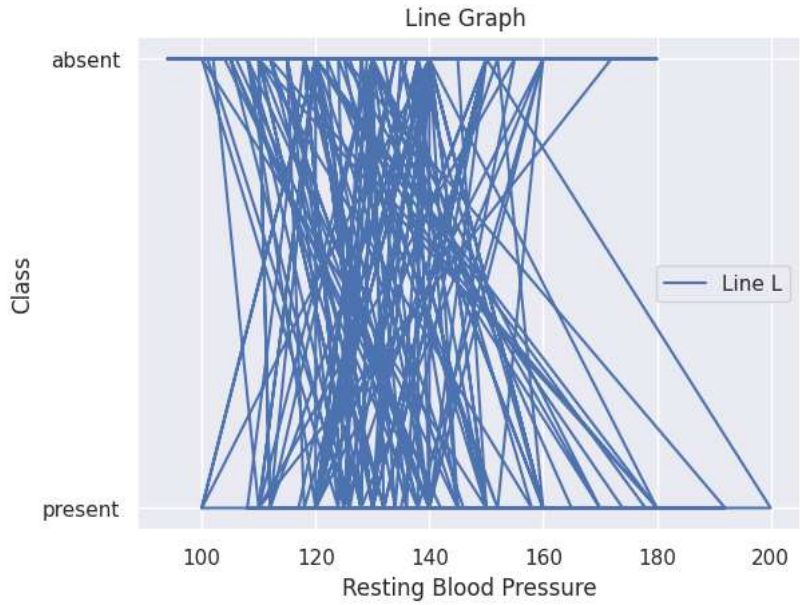
Data Visualisation using Matplotlib

```
import matplotlib.pyplot as plt
```

```
#No of patient who have heart disease and who don't
plt.hist(df['age'],color='green',edgecolor='white',bins=5)
plt.title('histrogram of Age')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```

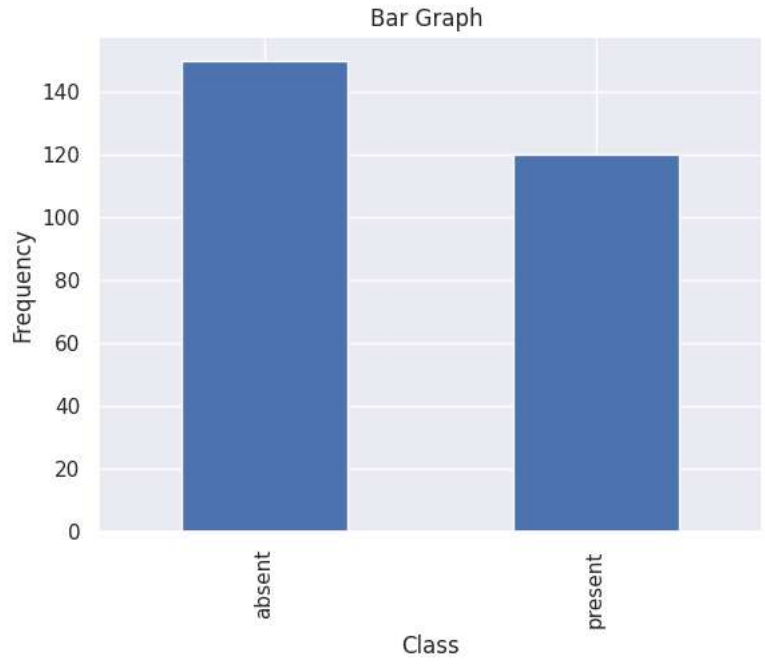



```
#most of our patients have resting blood pressure between 120 and 150
#line graph
plt.plot(df['resting_blood_pressure'],df['class'],label="Line L")
plt.plot()
plt.xlabel("Resting Blood Pressure")
plt.ylabel("Class")
plt.title("Line Graph ")
plt.legend()
plt.show()
```




```
df["class"].value_counts().plot(kind='bar')
plt.xlabel("Class")
plt.ylabel("Frequency")
plt.title("Bar Graph ")
```

```
Text(0.5, 1.0, 'Bar Graph ')
```

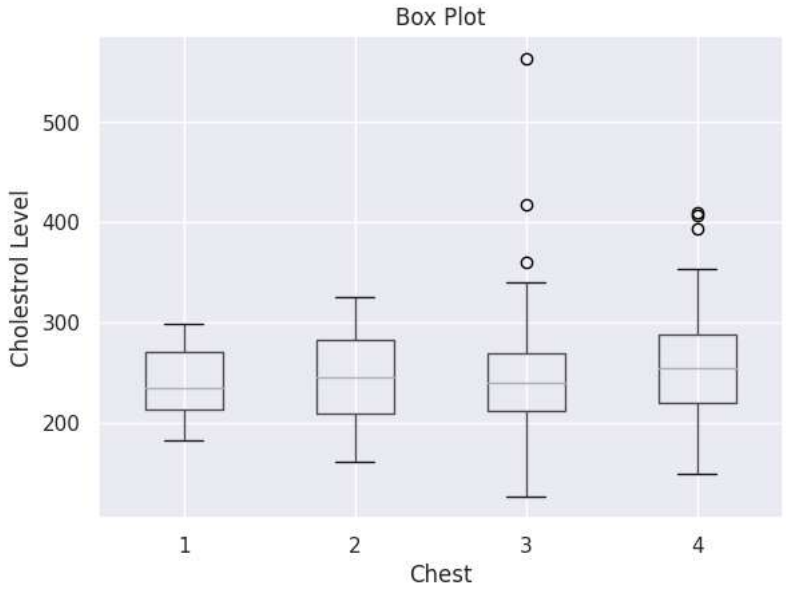



```
plt.scatter(df['serum_cholestoral'],df['age'],marker='*',color='r')
plt.scatter(df['resting_blood_pressure'],df['age'],marker='^',color='b')
plt.scatter(df['maximum_heart_rate_achieved'],df['age'],marker='.',color='g')
plt.title("Scatter Plot ")
plt.xlabel("Cholestrol/Blood Pressure/Heart Rate")
plt.ylabel("Age")
plt.legend()
plt.show()
```

```
WARNING:matplotlib.legend:No artists with labels found to put in legend.  Note that artists whose label
```


```
df.boxplot(column = 'serum_cholestoral', by = 'chest')
plt.suptitle('') # remove the automatic title
plt.title('Box Plot')
plt.ylabel('Cholestrol Level')
plt.xlabel('Chest')
```

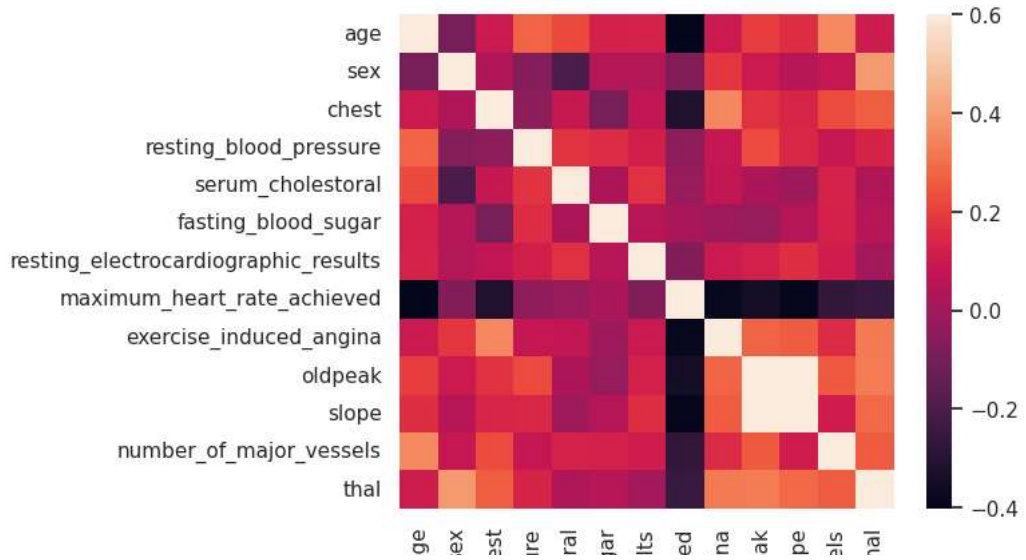
```
Text(0.5, 0, 'Chest')
```



Correlation Analysis

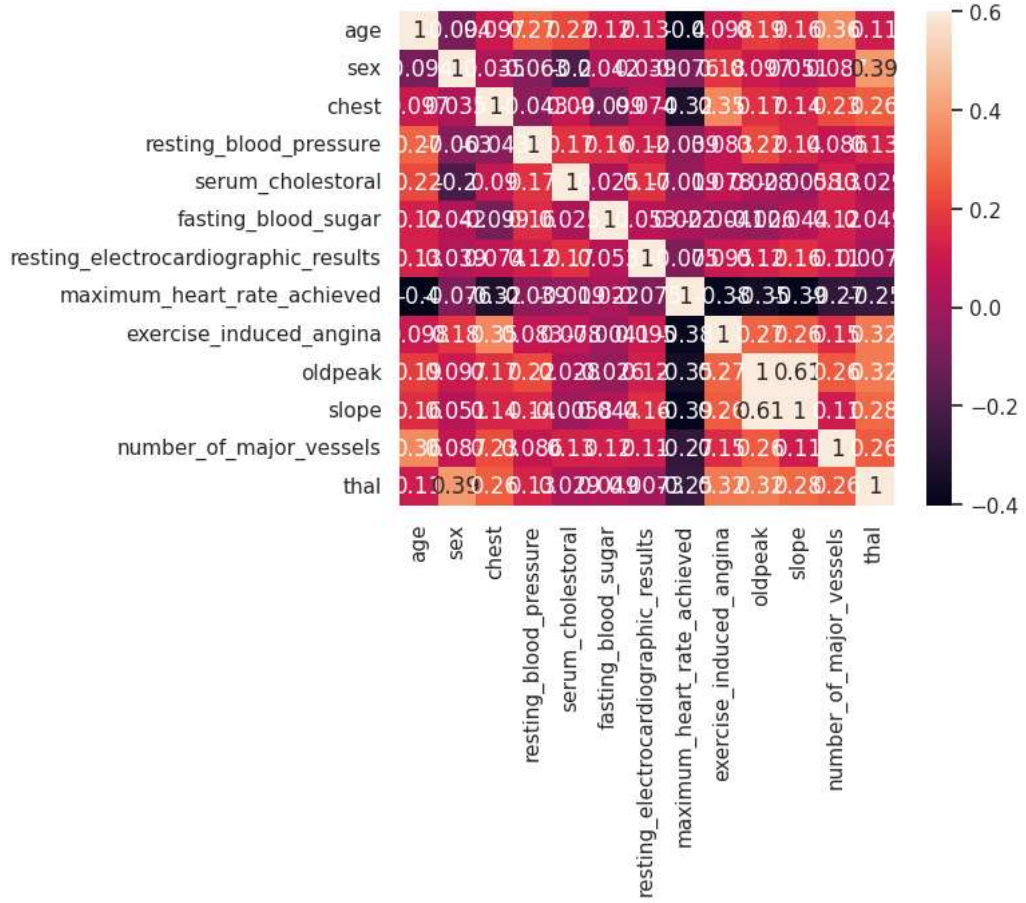
```
#correlation heatmap
#The lighter the shade,greater is the correlation
import seaborn as sns
corel=df.corr()
sns.heatmap(corel,vmax=0.6,square=True)
plt.show()
```

```
<ipython-input-445-859e8741a386>:4: FutureWarning: The default value of numeric_only in DataFrame.corr
  corel=df.corr()
```




```
corel=df.corr()
sns.heatmap(corel,vmax=0.6,square=True,annot=True)
plt.show()
```

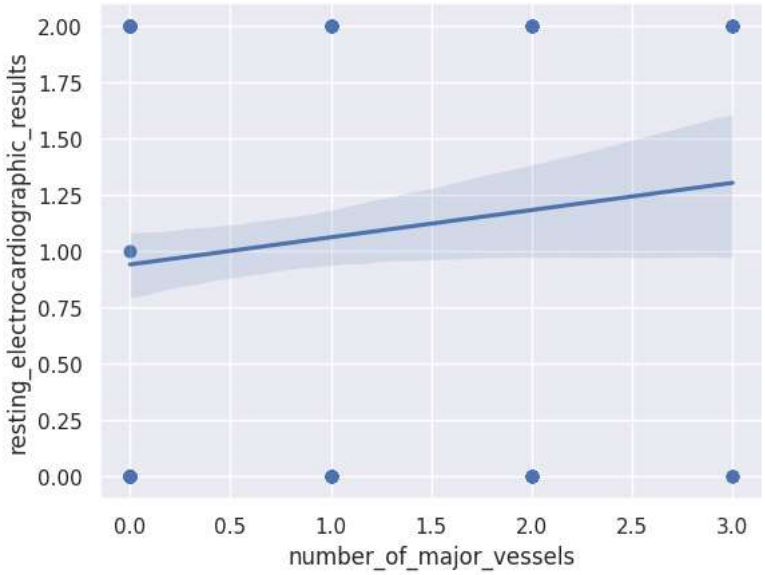
```
<ipython-input-446-dee615822d96>:1: FutureWarning: The default value of numeric_only in DataFrame.corr
  corel=df.corr()
```



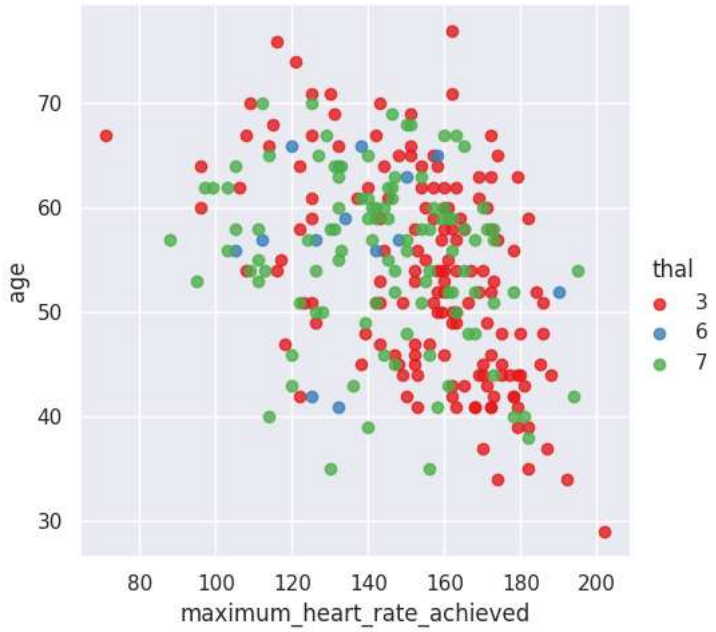
Data Visualisation using Seaborn

```
#regression plot
sns.set(style="darkgrid")
```

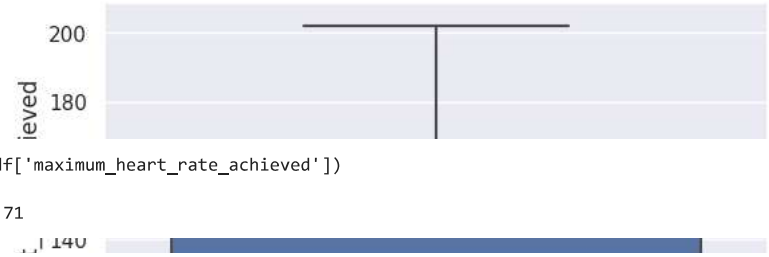
```
sns.regplot(x=df['number_of_major_vessels'],y=df['resting_electrocardiographic_results'])
plt.show()
```




```
#lm plot
sns.lmplot(x='maximum_heart_rate_achieved',y='age',data=df,fit_reg=False,hue='thal',legend=True,palette='Set1')
plt.show()
```




```
sns.boxplot(y=df['maximum_heart_rate_achieved'])
plt.show()
```

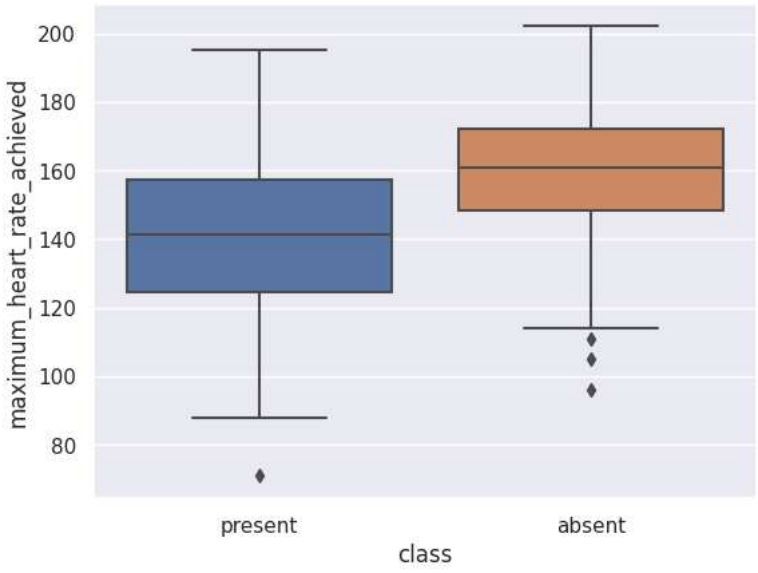



```
min(df['maximum_heart_rate_achieved'])
```

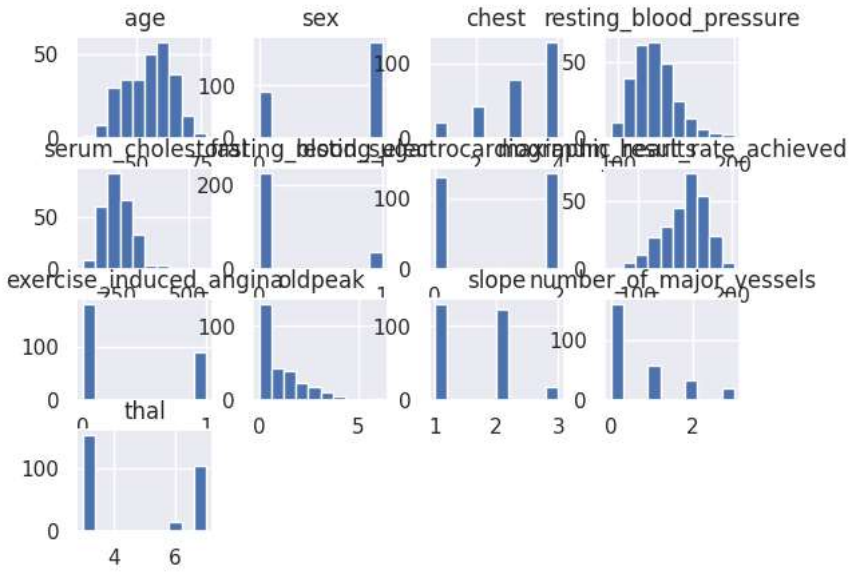
```
71
```



```
sns.boxplot(x=df['class'],y=df['maximum_heart_rate_achieved'])
plt.show()
```

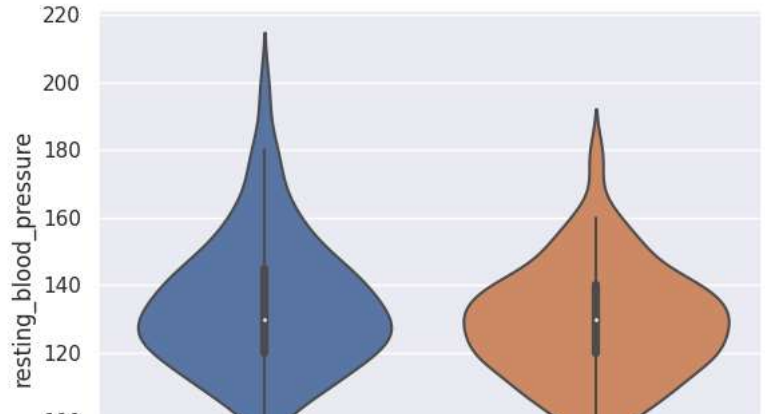



```
df.hist()
plt.show()
```




```
sns.violinplot(x=df['class'],y=df['resting_blood_pressure'])
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Prediction using Decision Tree

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
encoded=le.fit_transform(df['class'])
df['encoded_class']=encoded
print(df)
```

```
      age  sex  chest  resting_blood_pressure  serum_cholestoral  \
0      70    1      4                     130                322
1      67    0      3                     115                564
2      57    1      2                     124                261
3      64    1      4                     128                263
4      74    0      2                     120                269
..    ...  ...    ...                     ...                ...
265    52    1      3                     172                199
266    44    1      2                     120                263
267    56    0      2                     140                294
268    57    1      4                     140                192
269    67    1      4                     160                286

      fasting_blood_sugar  resting_electrocardiographic_results  \
0                       0                                    2
1                       0                                    2
2                       0                                    0
3                       0                                    0
4                       0                                    2
..                    ...                                  ...
265                     1                                    0
266                     0                                    0
267                     0                                    2
268                     0                                    0
269                     0                                    2

      maximum_heart_rate_achieved  exercise_induced_angina  oldpeak  slope  \
0                             109                        0      2.4      2
1                             160                        0      1.6      2
2                             141                        0      0.3      1
3                             105                        1      0.2      2
4                             121                        1      0.2      1
..                            ...                      ...      ...    ...
265                           162                        0      0.5      1
266                           173                        0      0.0      1
267                           153                        0      1.3      2
268                           148                        0      0.4      2
269                           108                        1      1.5      2

      number_of_major_vessels  thal    class  encoded_class
0                           3     3  present              1
1                           0     7   absent              0
2                           0     7  present              1
3                           1     7   absent              0
4                           1     3   absent              0
..                        ...   ...      ...            ...
265                         0     7   absent              0
266                         0     7   absent              0
267                         0     3   absent              0
268                         0     6   absent              0
269                         3     3  present              1

[270 rows x 15 columns]
```

```
df.columns
```

```
Index(['age', 'sex', 'chest', 'resting_blood_pressure', 'serum_cholestoral',
       'fasting_blood_sugar', 'resting_electrocardiographic_results',
       'maximum_heart_rate_achieved', 'exercise_induced_angina', 'oldpeak',
       'slope', 'number_of_major_vessels', 'thal', 'class', 'encoded_class'],
      dtype='object')
```

```
df.drop(['class'],axis='columns', inplace=True)
```

```
df.shape
```

```
(270, 14)
```

```
df.columns
```

```
Index(['age', 'sex', 'chest', 'resting_blood_pressure', 'serum_cholestoral',
       'fasting_blood_sugar', 'resting_electrocardiographic_results',
       'maximum_heart_rate_achieved', 'exercise_induced_angina', 'oldpeak',
       'slope', 'number_of_major_vessels', 'thal', 'encoded_class'],
      dtype='object')
```

```
Y=df[['encoded_class']]
X=df.drop(columns=['encoded_class'])
```

```
X
```

	age	sex	chest	resting_blood_pressure	serum_cholestoral	fasting_blood_sugar	resting_electroca
0	70	1	4	130	322	0	
1	67	0	3	115	564	0	
2	57	1	2	124	261	0	
3	64	1	4	128	263	0	
4	74	0	2	120	269	0	
...	...	...	...	...	...	...	
265	52	1	3	172	199	1	
266	44	1	2	120	263	0	
267	56	0	2	140	294	0	
268	57	1	4	140	192	0	
269	67	1	4	160	286	0	

270 rows × 13 columns

```
Y
```

	encoded_class
0	1
1	0
3	0
205	0

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.25,random_state=42)
```

```
x_train.shape
```

```
(202, 13)
```

```
x_test.shape
```

```
(68, 13)
```

```
#using decision tree model
from sklearn.tree import DecisionTreeClassifier
regressor=DecisionTreeClassifier(criterion='gini',max_depth=5,splitter='best')
X
regressor.fit(x_train,y_train)#training of classifier
```

```
DecisionTreeClassifier
DecisionTreeClassifier(max_depth=5)
```

```
y_test['Predicted']=regressor.predict(x_test)
```

```
y_test
```

	encoded_class	Predicted
30	1	0
116	1	1
79	0	0
127	0	0
196	0	1
...	...	...
10	0	1
233	1	1
66	0	0
253	0	0
225	0	1

68 rows × 2 columns

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

```
print(confusion_matrix(y_test['encoded_class'],y_test['Predicted'],))
```

```
[[30 10]
 [ 6 22]]
```

```
accuracy=accuracy_score(y_test['encoded_class'],y_test['Predicted'])
accuracy=np.round(accuracy,2)
print("Accuracy of the model is : ",accuracy)
```

```
Accuracy of the model is :  0.76
```

```
print(classification_report(y_test['encoded_class'],y_test['Predicted']))
```

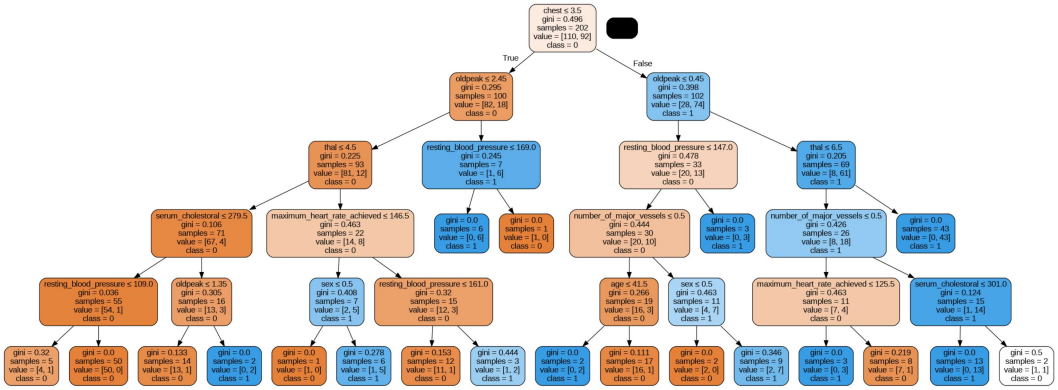
```
              precision    recall  f1-score   support

           0       0.83      0.75      0.79        40
```

```
                1       0.69      0.79      0.73        28

         accuracy                           0.76        68
        macro avg       0.76      0.77      0.76        68
     weighted avg       0.77      0.76      0.77        68
```

```
from six import StringIO
from IPython.display import Image
from sklearn.tree import export_graphviz
import pydotplus
feature_cols=['age', 'sex', 'chest', 'resting_blood_pressure', 'serum_cholestoral',
       'fasting_blood_sugar', 'resting_electrocardiographic_results',
       'maximum_heart_rate_achieved', 'exercise_induced_angina', 'oldpeak',
       'slope', 'number_of_major_vessels', 'thal']
dot_data=StringIO()
#regressor is the decision tree model
#filled= true so that different colours of boxes with different significance
#boxes have rounded corners
#feature names-names of each box , if we dont define it'll take 0,1,2 etc. as default
export_graphviz(regressor,out_file=dot_data,filled=True,rounded=True,special_characters=True,
                feature_names=feature_cols,class_names=['0','1'])
graph=pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png("tree.png")
Image(graph.create_png())
```




```
d1=X.iloc[16]
d1
```

```
age                                  46.0
sex                                   1.0
chest                                 4.0
resting_blood_pressure              140.0
serum_cholestoral                   311.0
fasting_blood_sugar                   0.0
resting_electrocardiographic_results  0.0
maximum_heart_rate_achieved         120.0
exercise_induced_angina               1.0
oldpeak                               1.8
slope                                 2.0
number_of_major_vessels               2.0
thal                                  7.0
Name: 16, dtype: float64
```

```
val_data=pd.DataFrame(d1)
val_data.transpose()
```

	age	sex	chest	resting_blood_pressure	serum_cholestoral	fasting_blood_sugar	resting_electroca
16	46.0	1.0	4.0	140.0	311.0	0.0	

```
#predicting class for a random record
y_pred=regressor.predict(val_data.transpose())
print("Prediction: ",y_pred)
```

```
Prediction:  [1]
```

```
df.iloc[16]
```

```
age                                  46.0
sex                                   1.0
chest                                 4.0
resting_blood_pressure              140.0
serum_cholestoral                   311.0
fasting_blood_sugar                   0.0
resting_electrocardiographic_results  0.0
maximum_heart_rate_achieved         120.0
exercise_induced_angina               1.0
oldpeak                               1.8
slope                                 2.0
number_of_major_vessels               2.0
thal                                  7.0
encoded_class                         1.0
Name: 16, dtype: float64
```