

In [1]:

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

In [2]:

```
%matplotlib inline
sns.set_style("whitegrid")
plt.style.use("fivethirtyeight")
```

In [3]:

```
df = pd.read_csv("D:/Users/Janhavi/TE/TE1/ML/Heart.csv")
```

In [4]:

```
df.head() #Returns the first 5 rows of the dataframe
```

Out[4]:

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpe
0	1	63	1	typical	145	233	1	2	150	0	;
1	2	67	1	asymptomatic	160	286	0	2	108	1	
2	3	67	1	asymptomatic	120	229	0	2	129	1	;
3	4	37	1	nonanginal	130	250	0	0	187	0	;
4	5	41	0	nontypical	130	204	0	2	172	0	

In [5]:

```
shape=df.shape #stores the number of rows and columns as a tuple (number of rows, number of columns)
```

In [6]:

```
print("Shape={}\nShape[0] x Shape[1]={}".format(shape,shape[0]*shape[1]))
```

```
Shape=(303, 15)
```

```
Shape[0] x Shape[1]=4545
```

In [7]:

```
df.info()      #used to print a concise summary of a DataFrame.
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 15 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   Unnamed: 0   303 non-null    int64
 1   Age          303 non-null    int64
 2   Sex          303 non-null    int64
 3   ChestPain    303 non-null    object
 4   RestBP       303 non-null    int64
 5   Chol         303 non-null    int64
 6   Fbs          303 non-null    int64
 7   RestECG      303 non-null    int64
 8   MaxHR        303 non-null    int64
 9   ExAng        303 non-null    int64
10   Oldpeak      303 non-null    float64
11   Slope        303 non-null    int64
12   Ca           299 non-null    float64
13   Thal         301 non-null    object
14   AHD          303 non-null    object
dtypes: float64(2), int64(10), object(3)
memory usage: 35.6+ KB
```

In [8]:

```
df.isnull()
#The isnull() method returns a DataFrame object where all the values are replaced with
a Boolean value True for NULL values, and otherwise False.
```

Out[8]:

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	O
0	False	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	False	
...	
298	False	False	False	False	False	False	False	False	False	False	
299	False	False	False	False	False	False	False	False	False	False	
300	False	False	False	False	False	False	False	False	False	False	
301	False	False	False	False	False	False	False	False	False	False	
302	False	False	False	False	False	False	False	False	False	False	

303 rows × 15 columns



In [9]:

```
df.notnull()
#notnull() function detects existing/ non-missing values in the dataframe.
```

Out[9]:

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldp
0	True	True	True	True	True	True	True	True	True	True	1
1	True	True	True	True	True	True	True	True	True	True	1
2	True	True	True	True	True	True	True	True	True	True	1
3	True	True	True	True	True	True	True	True	True	True	1
4	True	True	True	True	True	True	True	True	True	True	1
...
298	True	True	True	True	True	True	True	True	True	True	1
299	True	True	True	True	True	True	True	True	True	True	1
300	True	True	True	True	True	True	True	True	True	True	1
301	True	True	True	True	True	True	True	True	True	True	1
302	True	True	True	True	True	True	True	True	True	True	1

303 rows × 15 columns



In [10]:

```
df.notnull().sum()
#The function df. isnull(). sum(). sum () returns total number of missing values in the data set.
```

Out[10]:

```
Unnamed: 0    303
Age           303
Sex           303
ChestPain     303
RestBP        303
Chol          303
Fbs           303
RestECG       303
MaxHR         303
ExAng         303
Oldpeak       303
Slope         303
Ca            299
Thal          301
AHD           303
dtype: int64
```

In [11]:

```
df.dtypes      # This attribute returns a Series with the data type of each column.
```

Out[11]:

```
Unnamed: 0      int64
Age             int64
Sex             int64
ChestPain       object
RestBP          int64
Chol            int64
Fbs            int64
RestECG         int64
MaxHR           int64
ExAng           int64
Oldpeak         float64
Slope           int64
Ca             float64
Thal            object
AHD             object
dtype: object
```

In [12]:

```
df.all()
# all() method is used to check whether all the elements of a DataFrame are zero or not.
```

Out[12]:

```
Unnamed: 0      True
Age             True
Sex             False
ChestPain       True
RestBP          True
Chol            True
Fbs            False
RestECG         False
MaxHR           True
ExAng           False
Oldpeak         False
Slope           True
Ca             False
Thal            True
AHD             True
dtype: bool
```

In [13]:

```
(df==0).sum().sum()
```

Out[13]:

985

In [14]:

```
df['Age'].mean()
# To calculate the mean over the column called above 'Age'
```

Out[14]:

54.43894389438944

In [15]:

```
df=df[["Age","Sex","ChestPain","Chol"]] # To extract only Age, Sex, ChestPain, Chol.
```

In [16]:

```
print(df)
```

	Age	Sex	ChestPain	Chol
0	63	1	typical	233
1	67	1	asymptomatic	286
2	67	1	asymptomatic	229
3	37	1	nonanginal	250
4	41	0	nontypical	204
..
298	45	1	typical	264
299	68	1	asymptomatic	193
300	57	1	asymptomatic	131
301	57	0	nontypical	236
302	38	1	nonanginal	175

[303 rows x 4 columns]

In [17]:

```
df.shape #stores the number of rows and columns as a tuple
```

Out[17]:

(303, 4)

In []:

Assignment Part B

Through the diagnosis test I predicted 100 report **as** COVID positive, but only 45 of those

were actually positive. Total 50 people **in** my sample were actually COVID positive. I have

total 500 samples. based on table TP=45 FP=55 FN=50 TN=395 TOTAL=500

Actual

yes no

Predicted yes 45 55 =100

No 5 395 =400

50 450

Based on table

TP=45 FP=55 FN=5 TN=395 TOTAL=500

In [18]:

```
tp=45  
fp=55  
tn=395  
fn=5  
total=500
```

In [19]:

```
Acc=(tp+tn)/total  
print(Acc)
```

0.88

In [20]:

```
Precision=(tp)/(tp+fp)  
print(Precision)
```

0.45

In [21]:

```
Recall=(tp)/(tp+fn)  
print(Recall)
```

0.9

In [22]:

```
F_Score=2*((Precision*Recall)/(Precision+Recall))  
print(F_Score)
```

0.6