```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

```
from google.colab import files
uploaded = files.upload()
```

Choose Files heart.csv

- **heart.csv**(text/csv) - 35921 bytes, last modified: 1/29/2023 - 100% done

```
Saving heart.csv to heart.csv
```

```
df=pd.read_csv("heart.csv")
df
```

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	Max
0	40	M	ATA	140	289	0	Normal	1
1	49	F	NAP	160	180	0	Normal	1
2	37	M	ATA	130	283	0	ST	
3	48	F	ASY	138	214	0	Normal	1
4	54	M	NAP	150	195	0	Normal	1
...	...	...	...	...	...	...	...	
913	45	M	TA	110	264	0	Normal	1
914	68	M	ASY	144	193	1	Normal	1
915	57	M	ASY	130	131	0	Normal	1
916	57	F	ATA	130	236	0	LVH	1
917	38	M	NAP	138	175	0	Normal	1

918 rows × 12 columns

```
df.head()
```

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR
0	40	M	ATA	140	289	0	Normal	172
1	49	F	NAP	160	180	0	Normal	156
2	37	M	ATA	130	283	0	ST	98
3	48	F	ASY	138	214	0	Normal	108
4	54	M	NAP	150	195	0	Normal	122

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 918 entries, 0 to 917
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Age             918 non-null    int64  
 1   Sex             918 non-null    object 
 2   ChestPainType   918 non-null    object 
 3   RestingBP       918 non-null    int64  
 4   Cholesterol     918 non-null    int64  
 5   FastingBS       918 non-null    int64  
 6   RestingECG      918 non-null    object 
 7   MaxHR           918 non-null    int64  
 8   ExerciseAngina  918 non-null    object 
 9   Oldpeak         918 non-null    float64
 10  ST_Slope        918 non-null    object 
 11  HeartDisease    918 non-null    int64  
dtypes: float64(1), int64(6), object(5)
memory usage: 86.2+ KB
```

```
df.describe()
```

	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	He
count	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000	
mean	53.510893	132.396514	198.799564	0.233115	136.809368	0.887364	
std	9.432617	18.514154	109.384145	0.423046	25.460334	1.066570	
min	28.000000	0.000000	0.000000	0.000000	60.000000	-2.600000	
25%	47.000000	120.000000	173.250000	0.000000	120.000000	0.000000	
50%	54.000000	130.000000	223.000000	0.000000	138.000000	0.600000	
75%	60.000000	140.000000	267.000000	0.000000	156.000000	1.500000	
max	77.000000	200.000000	603.000000	1.000000	202.000000	6.200000	

```
df.Cholesterol = ( df.Cholesterol - df.Cholesterol.mean() ) / df.Cholesterol.std()
df.head(5)
```

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR
0	40	M	ATA	140	0.824621	0	Normal	172
1	49	F	NAP	160	-0.171867	0	Normal	156
2	37	M	ATA	130	0.769768	0	ST	98
3	48	F	ASY	138	0.138964	0	Normal	108
4	54	M	NAP	150	-0.034736	0	Normal	122

```
df[df['Cholesterol']>3]
```

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	Max
76	32	M	ASY	118	3.018723	0	Normal	1
149	54	M	ASY	130	3.695238	1	Normal	1
616	67	F	NAP	115	3.338696	0	LVH	1

```
df[df['Cholesterol']<-3]
```

Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR

```
df1=df[df.Cholesterol<=(df.Cholesterol<-3) | (df.Cholesterol>3)]
```

```
df1
```

```
df.Oldpeak = ( df.Oldpeak - df.Oldpeak.mean() ) / df.Oldpeak.std()
df.head(5)
```

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR
0	40	M	ATA	140	0.824621	0	Normal	172
1	49	F	NAP	160	-0.171867	0	Normal	156
2	37	M	ATA	130	0.769768	0	ST	98
3	48	F	ASY	138	0.138964	0	Normal	108
4	54	M	NAP	150	-0.034736	0	Normal	122

```
df[df['Oldpeak']>3]
```

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	Max
166	50	M	ASY	140	0.294379	0	ST	1
702	59	M	TA	178	0.650921	0	LVH	1
771	55	M	ASY	140	0.166390	0	Normal	1
791	51	M	ASY	140	0.906900	0	Normal	1
850	62	F	ASY	160	-0.318141	0	LVH	1
900	58	M	ASY	114	1.089741	0	ST	1

```
df[df['Oldpeak']<-3]
```

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	Max
324	46	M	ASY	100	-1.817444	1	ST	1

```
df2=df1[df1.Oldpeak<=(df1.Oldpeak<-3) | (df1.Oldpeak>3)]
```

```
df2
```

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	Max
4	54	M	NAP	150	-0.034736	0	Normal	1
16	38	M	ASY	110	-0.025594	0	Normal	1
29	51	M	ATA	125	-0.098731	0	Normal	1
31	56	M	NAP	130	-0.290715	0	Normal	1
34	43	F	ATA	150	-0.117015	0	Normal	1
...	...	...	...	...	...	...	...	
894	58	F	ASY	130	-0.016452	0	Normal	1
898	35	M	ATA	122	-0.062162	0	Normal	1
909	63	F	ASY	124	-0.016452	0	Normal	1
910	41	M	ATA	120	-0.382135	0	Normal	1
917	38	M	NAP	138	-0.217578	0	Normal	1

177 rows × 12 columns

```
df2.ChestPainType.unique()
```

```
array(['NAP', 'ASY', 'ATA', 'TA'], dtype=object)
```

```
df2.RestingECG.unique()
```

```
array(['Normal', 'ST', 'LVH'], dtype=object)
```

```
df2.ExerciseAngina.unique()
```

```
     array(['N', 'Y'], dtype=object)
```

```
df2.ST_Slope.unique()
```

```
     array(['Up', 'Flat', 'Down'], dtype=object)
```

```
df2.Sex.unique()
```

```
     array(['M', 'F'], dtype=object)
```

```
df2.Sex.replace(
    {
        'M': 0,
        'F': 1
    },
    inplace=True)
```

```
df2.ChestPainType.replace(
    {
        'NAP': 1,
        'ASY': 2,
        'ATA': 3,
        'TA' :4
    },
    inplace=True
)
```

```
df2.ExerciseAngina.replace(
    {
        'N': 0,
        'Y': 1
    },
    inplace=True)
```

```
df2.ST_Slope.replace(
    {
        'Down': 1,
        'Flat': 2,
        'Up': 3
    },
    inplace=True
)
```

```
df2.RestingECG.replace(
    {
        'Normal': 1,
        'ST': 2,
        'LVH': 3
    },
    inplace=True)
```

```
x = df2.drop("HeartDisease",axis='columns')
y = df2.HeartDisease
```

```
x
```

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	Max
4	54	0	1	150	-0.034736	0	1	1

```
y
```

```
    4      0
    16     1
    29     0
    31     0
    34     0
          ..
    894    0
    898    0
    909    1
    910    0
    917    0
    Name: HeartDisease, Length: 177, dtype: int64
```

910	41	0	2	120	0.282125	0	1	1

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

```
    array([[ 0.19644296, -0.42426407, -1.28614258, ..., -0.53160953,
            -0.06296298,  0.88794003],
           [-1.47923919, -0.42426407, -0.06877768, ..., -0.53160953,
            -0.06296298, -0.91855865],
           [-0.11774744, -0.42426407,  1.14858722, ..., -0.53160953,
            -0.06296298,  0.88794003],
           ...,
           [ 1.13901417,  2.3570226 , -0.06877768, ...,  1.88107989,
            -0.06296298, -0.91855865],
           [-1.16504878, -0.42426407,  1.14858722, ..., -0.53160953,
            -0.06296298,  0.88794003],
           [-1.47923919, -0.42426407, -1.28614258, ..., -0.53160953,
            -0.06296298,  0.88794003]])
```

```
from sklearn.model_selection import train_test_split

xtrain,xtest,ytrain,ytest = train_test_split(xscaled, y,test_size=0.2)
```

```
from sklearn.decomposition import PCA

pca = PCA()
xpca = pca.fit_transform(x)
xpca
```

```
    array([[ 1.30415148e+01,  1.87477591e+01,  2.69516887e+00, ...,
             7.54156084e-02, -1.79053529e-01,  1.83773666e-01],
           [-3.78464217e+01, -1.58041875e+01,  5.56287649e+00, ...,
             1.26315340e-01, -1.48044667e-01,  4.09994600e-01],
           [-1.31092521e+01, -2.72989491e+00, -1.61144093e+00, ...,
            -1.45076040e-02, -3.32061750e-01,  1.25078742e-01],
           ...,
           [-2.25756216e+00, -3.69848774e+00, -1.19807589e+01, ...,
             2.30924355e-01,  2.52047957e-01, -8.35758802e-01],
           [-5.16345116e+01, -3.26158803e+00,  1.35870583e+00, ...,
            -3.28638156e-02, -3.31232954e-01,  1.25232450e-01],
           [-4.11163271e+01,  1.27483487e+01,  8.25420482e+00, ...,
             2.37996940e-02, -2.03041889e-01,  1.69603307e-01]])
```

```
xtrain_pca, xtest_pca, ytrain, ytest = train_test_split(xpca, y, test_size=0.2
```

```
from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier()
rf.fit(xtrain_pca, ytrain)
rf.score(xtest_pca, ytest)
```

```
    0.8055555555555556
```

✓ 0s completed at 7:01 PM