In [1]:
```
import pandas as pd
df=pd.read_csv('https://raw.githubusercontent.com/codebasics/py/master/ML/18_PCA/Exercise/heart.csv')
df.head()
```

Out[1]:

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

In [2]:
```
df.shape
```

Out[2]:
```
(918, 12)
```

Treat Outliers

In [5]:
```
df[df.Cholesterol>(df.Cholesterol.mean()+3*df.Cholesterol.std())]
```

Out[5]:

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	
76	32	M	ASY	118	529	0	Normal	130	N	
149	54	M	ASY	130	603	1	Normal	125	Y	
616	67	F	NAP	115	564	0	LVH	160	N	

In [6]:
```
df.shape
```

Out[6]:
```
(918, 12)
```

In [8]:
```
df1=df[df.Cholesterol<=(df.Cholesterol.mean()+3*df.Cholesterol.std())]
df1.shape
```

Out[8]:
```
(915, 12)
```

In [9]:
```
df[df.MaxHR>(df.MaxHR.mean()+3*df.MaxHR.std())]
```

Out[9]:

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldp

In [10]:
```
df[df.FastingBS>(df.FastingBS.mean()+3*df.FastingBS.std())]
```

Out[10]:

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldp

In [12]:
```
df[df.Oldpeak>(df.Oldpeak.mean()+3*df.Oldpeak.std())]
```

Out[12]:

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	
166	50	M	ASY	140	231	0	ST	140	Y	
702	59	M	TA	178	270	0	LVH	145	N	
771	55	M	ASY	140	217	0	Normal	111	Y	
791	51	M	ASY	140	298	0	Normal	122	Y	
850	62	F	ASY	160	164	0	LVH	145	N	
900	58	M	ASY	114	318	0	ST	140	N	

In [18]:
```
df2=df1[df1.Oldpeak<=(df.Oldpeak.mean()+3*df1.Oldpeak.std())]
df2.shape
```

Out[18]:
```
(909, 12)
```

In [19]:
```
df[df.RestingBP>(df.RestingBP.mean()+3*df.RestingBP.std())]
```

Out[19]:

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	
109	39	M	ATA	190	241	0	Normal	106	N	
241	54	M	ASY	200	198	0	Normal	142	Y	
365	64	F	ASY	200	0	0	Normal	140	Y	
399	61	M	ASY	200	0	1	ST	70	N	
592	61	M	ASY	190	287	1	LVH	150	Y	
732	56	F	ASY	200	288	1	LVH	133	Y	
759	54	M	ATA	192	283	0	LVH	195	N	

In [21]:
```
df3=df2[df2.RestingBP<=(df.RestingBP.mean()+3*df2.RestingBP.std())]
df2.shape
```

Out[21]:
```
(909, 12)
```

In [22]:
```
df.ChestPainType.unique()
```

Out[22]:
```
array(['ATA', 'NAP', 'ASY', 'TA'], dtype=object)
```

In [23]:
```
df.RestingECG.unique()
```

Out[23]:
```
array(['Normal', 'ST', 'LVH'], dtype=object)
```

In [24]:
```
df.ExerciseAngina.unique()
```

Out[24]:
```
array(['N', 'Y'], dtype=object)
```

In [25]:
```
df.ST_Slope.unique()
```

Out[25]:
```
array(['Up', 'Flat', 'Down'], dtype=object)
```

In [26]:
```
df4=df3.copy()
df4.ExerciseAngina.replace(
    {
        'N':0,
        'Y':1
    },
     inplace=True)

df4.ST_Slope.replace(
    {
        'Down': 1,
        'Flat': 2,
        'Up': 3
    },
    inplace=True
)

df4.RestingECG.replace(
    {
        'Normal': 1,
        'ST': 2,
        'LVH': 3
    },
    inplace=True)

df4.head()
```

Out[26]:

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

In [28]:
```
df5=pd.get_dummies(df4, drop_first=True)
df5.head()
```

Out[28]:

	Age	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	Hea
0	40	140	289	0	1	172	0	0.0	3	
1	49	160	180	0	1	156	0	1.0	2	
2	37	130	283	0	2	98	0	0.0	3	
3	48	138	214	0	1	108	1	1.5	2	
4	54	150	195	0	1	122	0	0.0	3	

In [29]:
```
X = df5.drop("HeartDisease",axis='columns')
y = df5.HeartDisease

X.head()
```

Out[29]:

	Age	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	Sex
0	40	140	289	0	1	172	0	0.0	3	
1	49	160	180	0	1	156	0	1.0	2	
2	37	130	283	0	2	98	0	0.0	3	
3	48	138	214	0	1	108	1	1.5	2	
4	54	150	195	0	1	122	0	0.0	3	

In [30]:
```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_scaled
```

Out[30]:
```
array([[-1.42896269,  0.46083071,  0.85238015, ...,  2.06757196,
        -0.53547478, -0.22914788],
       [-0.47545956,  1.5925728 , -0.16132855, ..., -0.4836591 ,
         1.86750159, -0.22914788],
       [-1.74679706, -0.10495034,  0.79657967, ...,  2.06757196,
        -0.53547478, -0.22914788],
       ...,
       [ 0.37209878, -0.10495034, -0.61703246, ..., -0.4836591 ,
        -0.53547478, -0.22914788],
       [ 0.37209878, -0.10495034,  0.35947592, ...,  2.06757196,
        -0.53547478, -0.22914788],
       [-1.64085227,  0.3477225 , -0.20782894, ..., -0.4836591 ,
         1.86750159, -0.22914788]])
```

In [31]:
```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random
```

In [32]:
```
X_train.shape
```

Out[32]:
```
(721, 13)
```

In [33]:
```
X_test.shape
```

Out[33]:
```
(181, 13)
```

In [34]:
```
from sklearn.ensemble import RandomForestClassifier

model_rf = RandomForestClassifier()
model_rf.fit(X_train, y_train)
model_rf.score(X_test, y_test)
```

Out[34]:
```
0.8674033149171271
```

Use PCA to reduce dimensions

In [36]:
```
X
```

Out[36]:

	Age	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	S
0	40	140	289	0	1	172	0	0.0	3	
1	49	160	180	0	1	156	0	1.0	2	
2	37	130	283	0	2	98	0	0.0	3	
3	48	138	214	0	1	108	1	1.5	2	
4	54	150	195	0	1	122	0	0.0	3	
...	...	...	...	...	...	...	...	...	...	
913	45	110	264	0	1	132	0	1.2	2	
914	68	144	193	1	1	141	0	3.4	2	
915	57	130	131	0	1	115	1	1.2	2	
916	57	130	236	0	3	174	0	0.0	2	
917	38	138	175	0	1	173	0	0.0	3	

902 rows × 13 columns

In [37]:
```
from sklearn.decomposition import PCA

pca = PCA(0.95)
X_pca = pca.fit_transform(X)
X_pca
```

Out[37]:
```
array([[ 93.82465373, -29.40099458],
       [-15.58422331, -14.10909233],
       [ 83.29606634,  38.6867453 ],
       ...,
       [-67.57318721,  17.61319354],
       [ 40.70458237, -33.38750602],
       [-19.91368346, -37.29085722]])
```

In [38]:
```
X_train_pca, X_test_pca, y_train, y_test = train_test_split(X_pca, y, test_size=0.2,
```

In [39]:
```
from sklearn.ensemble import RandomForestClassifier

model_rf = RandomForestClassifier()
model_rf.fit(X_train_pca, y_train)
model_rf.score(X_test_pca, y_test)
```

Out[39]:
```
0.7182320441988951
```

In [ ]:
```

```