In [4]:

```
import numpy as np
import pandas as pd

from sklearn import svm
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
```

In [5]:

```
df= pd.read_csv(r"C:\Users\Divyam Chaturvedi\Downloads\Parkinson Disease Dataset.csv")
df.head(10)
```

Out[5]:

	name	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DI
0	phon_R01_S01_1	119.992	157.302	74.997	0.00784	0.00007	0.00370	0.00554	0.011
1	phon_R01_S01_2	122.400	148.650	113.819	0.00968	0.00008	0.00465	0.00696	0.013
2	phon_R01_S01_3	116.682	131.111	111.555	0.01050	0.00009	0.00544	0.00781	0.016
3	phon_R01_S01_4	116.676	137.871	111.366	0.00997	0.00009	0.00502	0.00698	0.015
4	phon_R01_S01_5	116.014	141.781	110.655	0.01284	0.00011	0.00655	0.00908	0.019
5	phon_R01_S01_6	120.552	131.162	113.787	0.00968	0.00008	0.00463	0.00750	0.013
6	phon_R01_S02_1	120.267	137.244	114.820	0.00333	0.00003	0.00155	0.00202	0.004
7	phon_R01_S02_2	107.332	113.840	104.315	0.00290	0.00003	0.00144	0.00182	0.004
8	phon_R01_S02_3	95.730	132.068	91.754	0.00551	0.00006	0.00293	0.00332	0.008
9	phon_R01_S02_4	95.056	120.103	91.226	0.00532	0.00006	0.00268	0.00332	0.008

10 rows × 24 columns

In [6]:

```
#Displaying all rows and columns
pd.set_option('display.max_columns',None)
pd.set_option('display.max_rows',None)
df
```

Out[6]:

	name	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Ji
0	phon_R01_S01_1	119.992	157.302	74.997	0.00784	0.000070	0.00370	0.00554	
1	phon_R01_S01_2	122.400	148.650	113.819	0.00968	0.000080	0.00465	0.00696	
2	phon_R01_S01_3	116.682	131.111	111.555	0.01050	0.000090	0.00544	0.00781	
3	phon_R01_S01_4	116.676	137.871	111.366	0.00997	0.000090	0.00502	0.00698	
4	phon_R01_S01_5	116.014	141.781	110.655	0.01284	0.000110	0.00655	0.00908	
5	phon_R01_S01_6	120.552	131.162	113.787	0.00968	0.000080	0.00463	0.00750	
6	phon_R01_S02_1	120.267	137.244	114.820	0.00333	0.000030	0.00155	0.00202	
7	phon_R01_S02_2	107.332	113.840	104.315	0.00290	0.000030	0.00144	0.00182	
8	phon_R01_S02_3	95.730	132.068	91.754	0.00551	0.000060	0.00293	0.00332	
9	phon_R01_S02_4	95.056	120.103	91.226	0.00532	0.000060	0.00268	0.00332	

In [7]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 195 entries, 0 to 194
Data columns (total 24 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   name              195 non-null    object 
 1   MDVP:Fo(Hz)       195 non-null    float64
 2   MDVP:Fhi(Hz)      195 non-null    float64
 3   MDVP:Flo(Hz)      195 non-null    float64
 4   MDVP:Jitter(%)    195 non-null    float64
 5   MDVP:Jitter(Abs)  195 non-null    float64
 6   MDVP:RAP          195 non-null    float64
 7   MDVP:PPQ          195 non-null    float64
 8   Jitter:DDP        195 non-null    float64
 9   MDVP:Shimmer      195 non-null    float64
 10  MDVP:Shimmer(dB)  195 non-null    float64
 11  Shimmer:APQ3      195 non-null    float64
 12  Shimmer:APQ5      195 non-null    float64
 13  MDVP:APQ          195 non-null    float64
 14  Shimmer:DDA       195 non-null    float64
 15  NHR               195 non-null    float64
 16  HNR               195 non-null    float64
 17  status            195 non-null    int64  
 18  RPDE              195 non-null    float64
 19  DFA               195 non-null    float64
 20  spread1           195 non-null    float64
 21  spread2           195 non-null    float64
 22  D2                195 non-null    float64
 23  PPE               195 non-null    float64
dtypes: float64(22), int64(1), object(1)
memory usage: 36.7+ KB
```

In [8]:
```
df.describe()
```

Out[8]:

	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Sh
count	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.
mean	154.228641	197.104918	116.324631	0.006220	0.000044	0.003306	0.003446	0.009920	0.
std	41.390065	91.491548	43.521413	0.004848	0.000035	0.002968	0.002759	0.008903	0.
min	88.333000	102.145000	65.476000	0.001680	0.000007	0.000680	0.000920	0.002040	0.
25%	117.572000	134.862500	84.291000	0.003460	0.000020	0.001660	0.001860	0.004985	0.
50%	148.790000	175.829000	104.315000	0.004940	0.000030	0.002500	0.002690	0.007490	0.
75%	182.769000	224.205500	140.018500	0.007365	0.000060	0.003835	0.003955	0.011505	0.
max	260.105000	592.030000	239.170000	0.033160	0.000260	0.021440	0.019580	0.064330	0.

In [9]:
```
df.shape
```

Out[9]: (195, 24)

In [10]:
```
df.isnull()
```

Out[10]:

	name	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	M
0	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	
5	False	False	False	False	False	False	False	False	False	
6	False	False	False	False	False	False	False	False	False	
7	False	False	False	False	False	False	False	False	False	
8	False	False	False	False	False	False	False	False	False	
9	False	False	False	False	False	False	False	False	False	

In [11]:
```
df.isnull().sum()
```

Out[11]:
```
name                0
MDVP:Fo(Hz)         0
MDVP:Fhi(Hz)        0
MDVP:Flo(Hz)        0
MDVP:Jitter(%)      0
MDVP:Jitter(Abs)    0
MDVP:RAP            0
MDVP:PPQ            0
Jitter:DDP          0
MDVP:Shimmer        0
MDVP:Shimmer(dB)    0
Shimmer:APQ3        0
Shimmer:APQ5        0
MDVP:APQ            0
Shimmer:DDA         0
NHR                 0
HNR                 0
status              0
RPDE                0
DFA                 0
spread1             0
spread2             0
D2                  0
PPE                 0
dtype: int64
```

In [12]:
```
df['status'].value_counts()
```

Out[12]:
```
1    147
0     48
Name: status, dtype: int64
```

In [13]:
```
df.groupby('status').mean()
```

```
C:\Users\Divyam Chaturvedi\AppData\Local\Temp\ipykernel_15564\4081209983.py:1: FutureWarning: The defaul
t value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version, numeric_only will d
efault to False. Either specify numeric_only or select only columns which should be valid for the functi
on.
  df.groupby('status').mean()
```

Out[13]:

status	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Sh
0	181.937771	223.636750	145.207292	0.003866	0.000023	0.001925	0.002056	0.005776	0.0
1	145.180762	188.441463	106.893558	0.006989	0.000051	0.003757	0.003900	0.011273	0.0

In [14]:
```
x=df.drop(columns=['name','status'],axis=1)
y=df['status']
y
```

Out[14]:
```
0      1
1      1
2      1
3      1
4      1
5      1
6      1
7      1
8      1
9      1
10     1
11     1
12     1
13     1
14     1
15     1
16     1
17     1
18     1
```

In [15]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=2)
print(x.shape,x_train.shape,x_test.shape)
```

```
(195, 22) (156, 22) (39, 22)
```

In [16]:
```
ss=StandardScaler()
ss.fit(x_train)
```

Out[16]:
```
▾ StandardScaler
StandardScaler()
```

In [17]:
```
x_train=ss.transform(x_train)
x_test=ss.transform(x_test)
```

In [18]:
```
print(x_train)
```

```
[[ 0.63239631 -0.02731081 -0.87985049 ... -0.97586547 -0.55160318
   0.07769494]
 [-1.05512719 -0.83337041 -0.9284778  ...  0.3981808  -0.61014073
   0.39291782]
 [ 0.02996187 -0.29531068 -1.12211107 ... -0.43937044 -0.62849605
  -0.50948408]
 ...
 [-0.9096785  -0.6637302  -0.160638   ...  1.22001022 -0.47404629
  -0.2159482 ]
 [-0.35977689  0.19731822 -0.79063679 ... -0.17896029 -0.47272835
   0.28181221]
 [ 1.01957066  0.19922317 -0.61914972 ... -0.716232    1.23632066
  -0.05829386]]
```

In [19]:
```
print(x_test)
```

```
[[-1.70008583e+00 -9.67968410e-01 -7.70130215e-01 -2.75000683e-01
   4.16156683e-01 -2.92615113e-01 -9.70869783e-02 -2.91621655e-01
  -4.94706656e-01 -4.90058396e-01 -5.32488171e-01 -4.26848854e-01
  -3.60251422e-01 -5.32484688e-01 -3.57189713e-01 -1.08840337e-01
   1.06963705e+00  1.05628304e+00  3.72180199e-01  1.94886208e+00
   3.66935071e-02  4.44314482e-01]
 [-1.39044095e+00 -9.29681132e-01 -7.37045677e-01  7.42068829e-01
   1.50451280e+00  8.54349819e-01  7.33639862e-01  8.53234751e-01
  -3.12538562e-03  3.01660094e-01  1.16511011e-01 -7.67595149e-02
  -2.23967413e-01  1.16829276e-01 -1.19644974e-01 -5.22790834e-01
   9.12650090e-01  1.31721995e+00  6.70118138e-01  4.74318608e-01
   1.42454868e-02  7.46859799e-01]
 [-1.35302065e+00 -6.29175292e-01 -7.29027225e-01  4.92094897e-01
   1.23242377e+00  4.52288742e-01  3.45291949e-01  4.53262231e-01
  -1.57435662e-01 -1.27992014e-01 -6.49095096e-02 -2.59345791e-01
  -2.60383383e-01 -6.52155416e-02  2.54927471e-01 -6.85306331e-01
   1.63423714e+00 -8.42551171e-01  2.43042190e+00  2.01645645e+00
   4.23263515e-01  1.70448737e+00]
 [ 1.04170416e+00  2.17641374e-01  6.81254572e-01 -5.21158220e-01
```

In [20]:
```
model=svm.SVC(kernel='linear')
```

In [21]:
```
model.fit(x_train,y_train)
```

Out[21]:
```
SVC
SVC(kernel='linear')
```

In [22]:
```
x_train_prediction=model.predict(x_train)
train_data_accuracy=accuracy_score(y_train,x_train_prediction)
```

In [23]:
```
print("accuracy of training data:",train_data_accuracy)
```

```
accuracy of training data: 0.8846153846153846
```

In [24]:
```
x_test_prediction=model.predict(x_test)
test_data_accuracy=accuracy_score(y_test,x_test_prediction)
```

In [25]:
```
print("accuracy of test data: ",test_data_accuracy)
```

```
accuracy of test data:  0.8717948717948718
```

In [26]:
```
input_data=(120.267,137.244,114.82,0.00333,0.00003,0.00155,0.00202,0.00466,0.01608,0.14,0.00779,0.00937,0
input_data_np=np.asarray(input_data)
input_data_re = input_data_np.reshape(1,-1)
s_data=ss.transform(input_data_re)
prediction=model.predict(s_data)
print(prediction)
if (prediction[0]==0):
    print("Negative,No Parkinsons's Found")
else :
    print("Positive,Parkinsons's Found")
```

```
[1]
Positive,Parkinsons's Found

C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py:439: UserWarning: X does not have valid featu
re names, but StandardScaler was fitted with feature names
  warnings.warn(
```