In [21]:

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score,confusion_matrix
```

In [3]:

```
data=pd.read_csv('diabetes.csv')
data.head()
```

Out[3]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunctio
0	6	148	72	35	0	33.6	0.62
1	1	85	66	29	0	26.6	0.35
2	8	183	64	0	0	23.3	0.67
3	1	89	66	23	94	28.1	0.16
4	0	137	40	35	168	43.1	2.28

In [11]:

```
data_n=data.iloc[:,:-1]
data_new=np.array(data_n)
data_new
```

Out[11]:

```
array([[  6.   , 148.   ,  72.   , ...,  33.6  ,   0.627,  50.   ],
       [  1.   ,  85.   ,  66.   , ...,  26.6  ,   0.351,  31.   ],
       [  8.   , 183.   ,  64.   , ...,  23.3  ,   0.672,  32.   ],
       ...,
       [  5.   , 121.   ,  72.   , ...,  26.2  ,   0.245,  30.   ],
       [  1.   , 126.   ,  60.   , ...,  30.1  ,   0.349,  47.   ],
       [  1.   ,  93.   ,  70.   , ...,  30.4  ,   0.315,  23.   ]])
```

In [13]:

```
covMat=pd.DataFrame(data=np.cov(data_new,rowvar=False))
covMat
```

Out[13]:

	0	1	2	3	4	5	6	
0	11.354056	13.947131	9.214538	-4.390041	-28.555231	0.469774	-0.037426	21.57
1	13.947131	1022.248314	94.430956	29.239183	1220.935799	55.726987	1.454875	99.08
2	9.214538	94.430956	374.647271	64.029396	198.378412	43.004695	0.264638	54.52
3	-4.390041	29.239183	64.029396	254.473245	802.979941	49.373869	0.972136	-21.38
4	-28.555231	1220.935799	198.378412	802.979941	13281.180078	179.775172	7.066681	-57.14
5	0.469774	55.726987	43.004695	49.373869	179.775172	62.159984	0.367405	3.36
6	-0.037426	1.454875	0.264638	0.972136	7.066681	0.367405	0.109779	0.13
7	21.570620	99.082805	54.523453	-21.381023	-57.143290	3.360330	0.130772	138.30

In [14]:

```
eignV, eignVe=np.linalg.eig(covMat)
```

In [15]:

```
eignV
```

Out[15]:

```
array([1.34565730e+04, 9.32760132e+02, 3.90577831e+02, 1.98182691e+02,
       1.12689115e+02, 4.58294431e+01, 7.76070899e+00, 1.02871018e-01])
```

In [16]:

```
eignVe
```

Out[16]:

```
array([[-2.02176587e-03,  2.26488861e-02, -2.24649003e-02,
        -4.90459604e-02, -1.51612874e-01, -5.04730888e-03,
         9.86672995e-01, -6.10123250e-03],
       [ 9.78115765e-02,  9.72210040e-01,  1.43428710e-01,
         1.19830016e-01,  8.79407680e-02,  5.07391813e-02,
         8.83426114e-04,  8.25459539e-04],
       [ 1.60930503e-02,  1.41909330e-01, -9.22467192e-01,
        -2.62742788e-01,  2.32165009e-01,  7.56365525e-02,
        -1.22975947e-03, -5.20865450e-04],
       [ 6.07566861e-02, -5.78614699e-02, -3.07013055e-01,
         8.84369380e-01, -2.59973487e-01,  2.21363068e-01,
        -3.76444746e-04,  2.54871909e-03],
       [ 9.93110844e-01, -9.46266913e-02,  2.09773019e-02,
        -6.55503615e-02,  1.72312241e-04, -6.13326472e-03,
         1.42307394e-03,  2.68965921e-04],
       [ 1.40108085e-02,  4.69729766e-02, -1.32444542e-01,
         1.92801728e-01, -2.14744823e-02, -9.70776708e-01,
        -2.73046214e-03,  2.67341863e-03],
       [ 5.37167919e-04,  8.16804621e-04, -6.39983017e-04,
         2.69908637e-03, -1.64080684e-03, -2.02903702e-03,
        -6.34402965e-03, -9.99972146e-01],
       [-3.56474430e-03,  1.40168181e-01, -1.25454310e-01,
        -3.01024330e-01, -9.20504903e-01, -1.51133239e-02,
        -1.62555343e-01,  1.95271966e-03]])
```

In [55]:

```
totalSum=sum(eignV)
percent=[(i/totalSum)for i in sorted(eignV,reverse=True)]
percent
```

Out[55]:

```
[0.888546634575624,
 0.06159078374587651,
 0.025790118917661545,
 0.013086137415140801,
 0.00744093863988756７,
 0.0030261491882255707,
 0.0005124448745698516,
 6.79264301407264e-06]
```

In [47]:

```
pca4c=eignVe[:,:4]
pca4c
```

Out[47]:

```
array([[-2.02176587e-03,  2.26488861e-02, -2.24649003e-02,
        -4.90459604e-02],
       [ 9.78115765e-02,  9.72210040e-01,  1.43428710e-01,
         1.19830016e-01],
       [ 1.60930503e-02,  1.41909330e-01, -9.22467192e-01,
        -2.62742788e-01],
       [ 6.07566861e-02, -5.78614699e-02, -3.07013055e-01,
         8.84369380e-01],
       [ 9.93110844e-01, -9.46266913e-02,  2.09773019e-02,
        -6.55503615e-02],
       [ 1.40108085e-02,  4.69729766e-02, -1.32444542e-01,
         1.92801728e-01],
       [ 5.37167919e-04,  8.16804621e-04, -6.39983017e-04,
         2.69908637e-03],
       [-3.56474430e-03,  1.40168181e-01, -1.25454310e-01,
        -3.01024330e-01]])
```

In [48]:

```
pca4cT=np.transpose(pca4c)
dataT=np.transpose(data_new)
newdata=np.matmul(pca4cT,dataT)
newDataSet=np.transpose(newdata)
newDataSet
```

Out[48]:

```
array([[ 18.04202911, 160.80251286, -66.79368843,  20.90462808],
       [ 11.39841643,  95.943517  , -65.02957086,  14.23991254],
       [ 19.12604059, 192.75822668, -40.07109135,  -0.41769765],
       ...,
       [125.86988229, 121.47506542, -61.12076394,   4.35691332],
       [ 13.54218971, 139.03774998, -47.18163558,  -9.05890071],
       [ 12.4485343 , 103.23023416, -67.6856678 ,  19.05706431]])
```

In [49]:

```
x=newDataSet
y=data.iloc[:,8]
```

In [50]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

In [51]:

```
dtc=DecisionTreeClassifier()
dtc.fit(x_train,y_train)
```

Out[51]:

```
DecisionTreeClassifier()
```

In [52]:

```
y_predict=dtc.predict(x_test)
```

In [53]:

```
accuracy_score(y_test,y_predict)
```

Out[53]:

```
0.6493506493506493
```

In [54]:

```
confusion_matrix(y_test,y_predict)
```

Out[54]:

```
array([[109,  45],
       [ 36,  41]], dtype=int64)
```

In [ ]:

```
```