In [1]:

```
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 [2]:

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

Out[2]:

	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 [9]:

```
data_new=data.iloc[:,:-1]
data_new
```

Out[9]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunct
0	6	148	72	35	0	33.6	0.
1	1	85	66	29	0	26.6	0.
2	8	183	64	0	0	23.3	0.
3	1	89	66	23	94	28.1	0.
4	0	137	40	35	168	43.1	2.
...	...	...	...	...	...	...	
763	10	101	76	48	180	32.9	0.
764	2	122	70	27	0	36.8	0.
765	5	121	72	23	112	26.2	0.
766	1	126	60	0	0	30.1	0.
767	1	93	70	31	0	30.4	0.

768 rows × 8 columns

In [18]:

```
def sc(data):
    sdc=(data-np.mean(data,axis=0)) / np.std(data,axis=0)
    return sdc
scald=sc(data_new)
scald
```

Out[18]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedi
0	0.639947	0.848324	0.149641	0.907270	-0.692891	0.204013	
1	-0.844885	-1.123396	-0.160546	0.530902	-0.692891	-0.684422	
2	1.233880	1.943724	-0.263941	-1.288212	-0.692891	-1.103255	
3	-0.844885	-0.998208	-0.160546	0.154533	0.123302	-0.494043	
4	-1.141852	0.504055	-1.504687	0.907270	0.765836	1.409746	
...	...	...	...	...	...	...	
763	1.827813	-0.622642	0.356432	1.722735	0.870031	0.115169	
764	-0.547919	0.034598	0.046245	0.405445	-0.692891	0.610154	
765	0.342981	0.003301	0.149641	0.154533	0.279594	-0.735190	
766	-0.844885	0.159787	-0.470732	-1.288212	-0.692891	-0.240205	
767	-0.844885	-0.873019	0.046245	0.656358	-0.692891	-0.202129	

768 rows × 8 columns

In [19]:

```
def covmat(data):
    #print(data.shape[0])
    covariance=data.T.dot(data)/data.shape[0]
    return covariance
cov_mat=covmat(scald)
cov_mat
```

Out[19]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	
Pregnancies	1.000000	0.129459	0.141282	-0.081672	-0.073535	0.0
Glucose	0.129459	1.000000	0.152590	0.057328	0.331357	0.2
BloodPressure	0.141282	0.152590	1.000000	0.207371	0.088933	0.2
SkinThickness	-0.081672	0.057328	0.207371	1.000000	0.436783	0.3
Insulin	-0.073535	0.331357	0.088933	0.436783	1.000000	0.1
BMI	0.017683	0.221071	0.281805	0.392573	0.197859	1.0
DiabetesPedigreeFunction	-0.033523	0.137337	0.041265	0.183928	0.185071	0.1
Age	0.544341	0.263514	0.239528	-0.113970	-0.042163	0.0

In [20]:

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

In [21]:

```
eignV
```

Out[21]:

```
array([2.09437995, 1.73121014, 0.41981618, 0.40446205, 0.68262839,
       0.76234439, 0.87552904, 1.02962987])
```

In [22]:

```
eignVe
```

Out[22]:

```
array([[-0.1284321 , -0.59378583, -0.58879003,  0.11784098, -0.19359817,
         0.47560573, -0.08069115,  0.01308692],
       [-0.39308257, -0.17402908, -0.06015291,  0.45035526, -0.09416176,
        -0.46632804,  0.40432871, -0.46792282],
       [-0.36000261, -0.18389207, -0.19211793, -0.01129554,  0.6341159 ,
        -0.32795306, -0.05598649,  0.53549442],
       [-0.43982428,  0.33196534,  0.28221253,  0.5662838 , -0.00958944,
         0.48786206, -0.03797608,  0.2376738 ],
       [-0.43502617,  0.25078106, -0.13200992, -0.54862138,  0.27065061,
         0.34693481,  0.34994376, -0.33670893],
       [-0.45194134,  0.1009598 , -0.03536644, -0.34151764, -0.68537218,
        -0.25320376, -0.05364595,  0.36186463],
       [-0.27061144,  0.122069  , -0.08609107, -0.00825873,  0.08578409,
        -0.11981049, -0.8336801 , -0.43318905],
       [-0.19802707, -0.62058853,  0.71208542, -0.21166198,  0.03335717,
         0.10928996, -0.0712006 , -0.07524755]])
```

In [23]:

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

Out[23]:

```
[0.2617974931611004,
 0.21640126757746536,
 0.12870373364801915,
 0.10944113047600441,
 0.09529304819389635,
 0.08532854849331149,
 0.05247702246321915,
 0.050557755986983685]
```

In [24]:

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

Out[24]:

```
array([[-0.1284321 , -0.59378583, -0.58879003,  0.11784098],
       [-0.39308257, -0.17402908, -0.06015291,  0.45035526],
       [-0.36000261, -0.18389207, -0.19211793, -0.01129554],
       [-0.43982428,  0.33196534,  0.28221253,  0.5662838 ],
       [-0.43502617,  0.25078106, -0.13200992, -0.54862138],
       [-0.45194134,  0.1009598 , -0.03536644, -0.34151764],
       [-0.27061144,  0.122069  , -0.08609107, -0.00825873],
       [-0.19802707, -0.62058853,  0.71208542, -0.21166198]])
```

In [25]:

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

Out[25]:

	0	1	2	3
0	-125.517107	-58.501100	17.971556	64.303008
1	-88.310990	-34.406006	10.906275	38.425973
2	-113.050685	-65.791139	-6.109008	67.898731
3	-126.784684	-7.185631	-10.594607	-13.134855
4	-183.363387	6.703578	-6.448345	-32.824762
...	...	...	...	...
763	-195.152985	-12.170291	6.903213	-50.336444
764	-107.358556	-39.327566	3.550798	51.392468
765	-150.812204	-17.486597	-11.934297	-9.451800
766	-94.262143	-59.641143	12.678307	35.954195
767	-93.898728	-30.525916	4.393029	43.512027

768 rows × 4 columns

In [26]:

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

In [27]:

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

In [28]:

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

Out[28]:

```
DecisionTreeClassifier()
```

In [29]:

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

In [30]:

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

Out[30]:

```
0.696969696969697
```

In [31]:

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

Out[31]:

```
array([[124,  36],
       [ 34,  37]], dtype=int64)
```

In [ ]:

```

```