In [1]:
```
#01.Load the dataset
```

In [7]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
```

In [10]:
```
df=pd.read_csv('grapes_new.csv')
```

In [11]:
```
df.head()
```

Out[11]:

	COLOR_INTENSITY	SHAPE	RIPENESS_PER	ALCOHOL_PER	MALIC_ACID_PER	ASH	ALCALINITY_OF
0	120.53	elliptical	16.73	14.23	1.71	2.43	
1	112.57	elliptical	20.30	13.20	1.78	2.14	
2	120.78	elliptical	20.81	13.16	2.36	2.67	
3	134.17	elliptical	19.47	14.37	1.95	2.50	
4	112.19	round	17.57	13.24	2.59	2.87	

5 rows × 21 columns

In [16]:
```
#02.Seperate the input and output
```

In [19]:
```
x=df[['COLOR_INTENSITY']].values
y=df['SHAPE']
```

In [20]:
```
x
```

Out[20]:
```
array([[120.53],
       [112.57],
       [120.78],
       [134.17],
       [112.19],
       [127.54],
       [118.07],
       [116.8 ],
       [117.75],
       [130.51],
       [121.22],
       [116.49],
       [120.28],
       [119.01],
       [132.27],
       [131.01],
       [124.06],
       [126.59],
       [139.85],
       [120.91],
       [120.28],
       [112.7 ],
       [133.22],
       [138.08],
       [117.12],
       [120.59],
       [113.33],
       [108.91],
       [109.73],
       [107.14],
       [107.52],
       [115.23],
       [109.86],
       [113.33],
       [114.59],
       [120.91],
       [128.48],
       [109.16],
       [119.01],
       [111.44],
       [117.12],
       [113.96],
       [111.75],
       [108.28],
       [117.12],
       [123.62],
       [111.94],
       [119.2 ],
       [112.45],
       [116.74],
       [118.  ],
       [115.86],
       [123.43],
       [124.06],
       [141.11],
       [130.38],
       [120.28],
       [129.43],
       [124.7 ],
       [121.85],
```

```
       [124.38],
       [125.2 ],
       [122.8 ],
       [127.85],
       [110.81],
       [119.01],
       [120.91],
       [116.49],
       [119.33],
       [129.75],
       [109.23],
       [116.49],
       [120.91],
       [115.98],
       [113.96],
       [120.28],
       [112.38],
       [112.7 ],
       [136.76],
       [110.17],
       [104.18],
       [106.26],
       [108.53],
       [106.07],
       [105.19],
       [108.91],
       [115.86],
       [133.22],
       [138.08],
       [144.27],
       [139.22],
       [153.11],
       [129.75],
       [151.34],
       [132.9 ],
       [134.8 ],
       [141.81],
       [132.27],
       [167.  ],
       [159.11],
       [133.22],
       [122.04],
       [120.15],
       [118.26],
       [145.41],
       [126.72],
       [152.35],
       [149.7 ],
       [139.6 ],
       [138.59],
       [119.65],
       [147.43],
       [146.16],
       [133.54],
       [131.01],
       [149.32],
       [143.64],
       [143.01],
       [ 97.23],
       [105.56],
```

```
       [121.22],
       [108.91],
       [113.02],
       [103.54],
       [113.96],
       [118.38],
       [114.47],
       [104.93],
       [102.91],
       [104.18],
       [106.26],
       [108.53],
       [106.07],
       [105.19],
       [108.91],
       [113.96],
       [101.65],
       [106.39],
       [101.15],
       [100.7 ],
       [109.54],
       [ 98.81],
       [115.23],
       [104.18],
       [101.46],
       [100.39],
       [101.33],
       [102.6 ],
       [ 95.9 ],
       [100.07],
       [107.65],
       [104.18],
       [ 98.49],
       [105.44],
       [101.33],
       [100.7 ],
       [103.23],
       [113.33],
       [ 99.44],
       [105.75],
       [100.39],
       [102.6 ],
       [ 97.92],
       [103.48],
       [101.97],
       [106.39],
       [105.75],
       [101.97],
       [101.65],
       [103.23],
       [ 97.55],
       [108.91],
       [104.37],
       [103.23],
       [ 96.91],
       [ 97.23],
       [ 97.92],
       [106.39],
       [ 93.  ],
       [105.44],
```

```
       [122.8 ],
       [ 98.05],
       [127.54],
       [118.07],
       [116.8 ],
       [117.75],
       [130.51],
       [101.33],
       [102.6 ],
       [102.34],
       [109.8 ],
       [103.86],
       [ 98.3 ],
       [101.33]])
```

In [21]:
```
y
```

Out[21]:
```
0      elliptical
1      elliptical
2      elliptical
3      elliptical
4           round
          ...
189    elliptical
190    elliptical
191         round
192    elliptical
193    elliptical
Name: SHAPE, Length: 194, dtype: object
```

In [22]:
```
#03.Data cleaning
```

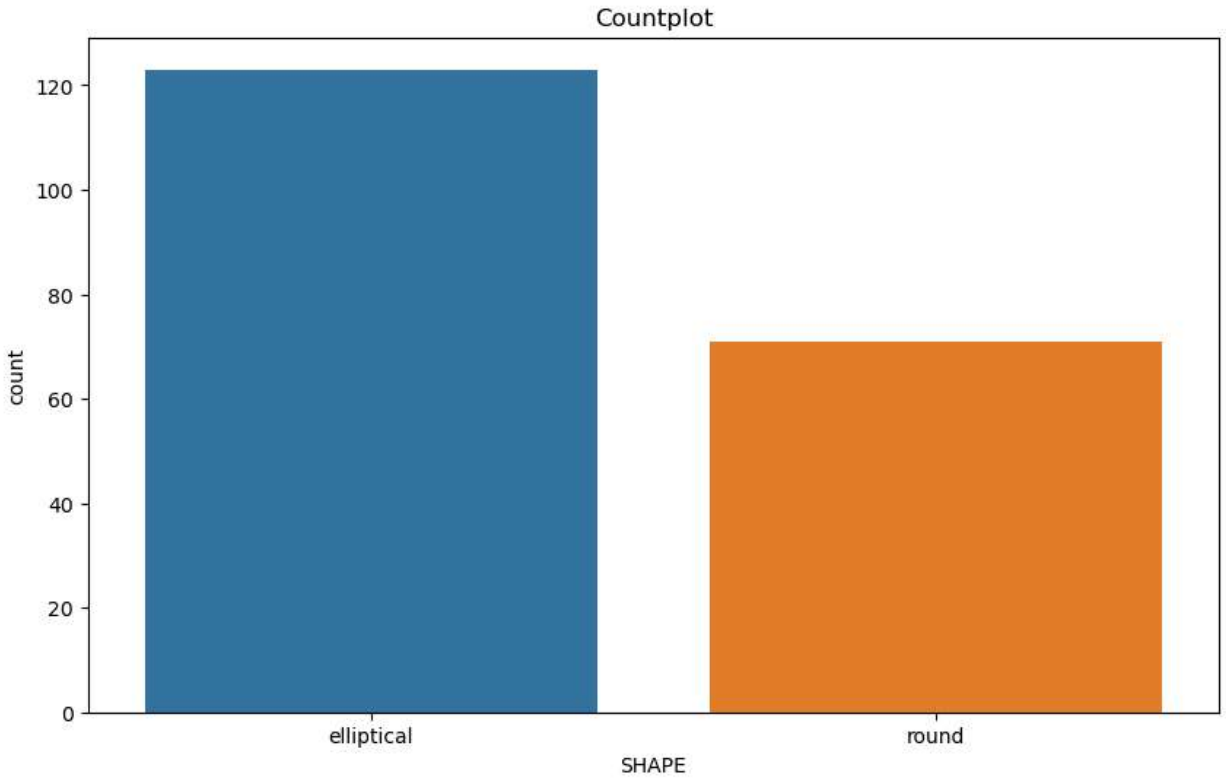
In [23]:
```
print(df.isnull().sum())  ###checking null values
```

```
COLOR_INTENSITY      0
SHAPE                0
RIPENESS_PER         2
ALCOHOL_PER          3
MALIC_ACID_PER       0
ASH                  0
ALCALINITY_OF_ASH    6
MAGNESIUM_MEASURE    0
TOTAL_PHENOLS        0
FLAVANOIDS           0
PROANTHOCYANINS      0
HUE                  2
PROLINE              0
TEMPERATURE          0
AVG_HUMIDITY         0
FERT_NITRO_PER       0
WATER_O2_PER         4
WATER_PH             0
FER_P2O5_PER         0
SOIL_TYPE            0
CLASS                0
dtype: int64
```

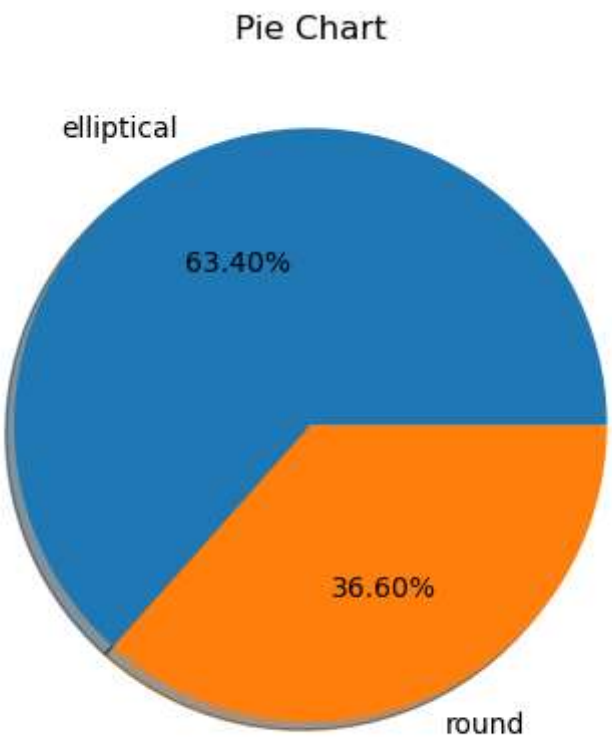
In [26]:
```
#04Explore the output variable through countplot,pie chart
import seaborn as sns
```

In [28]:
```
#Countplot
plt.figure(figsize=(10,6))
sns.countplot(x=y)
plt.title('Countplot')
```

Out[28]:
```
Text(0.5, 1.0, 'Countplot')
```



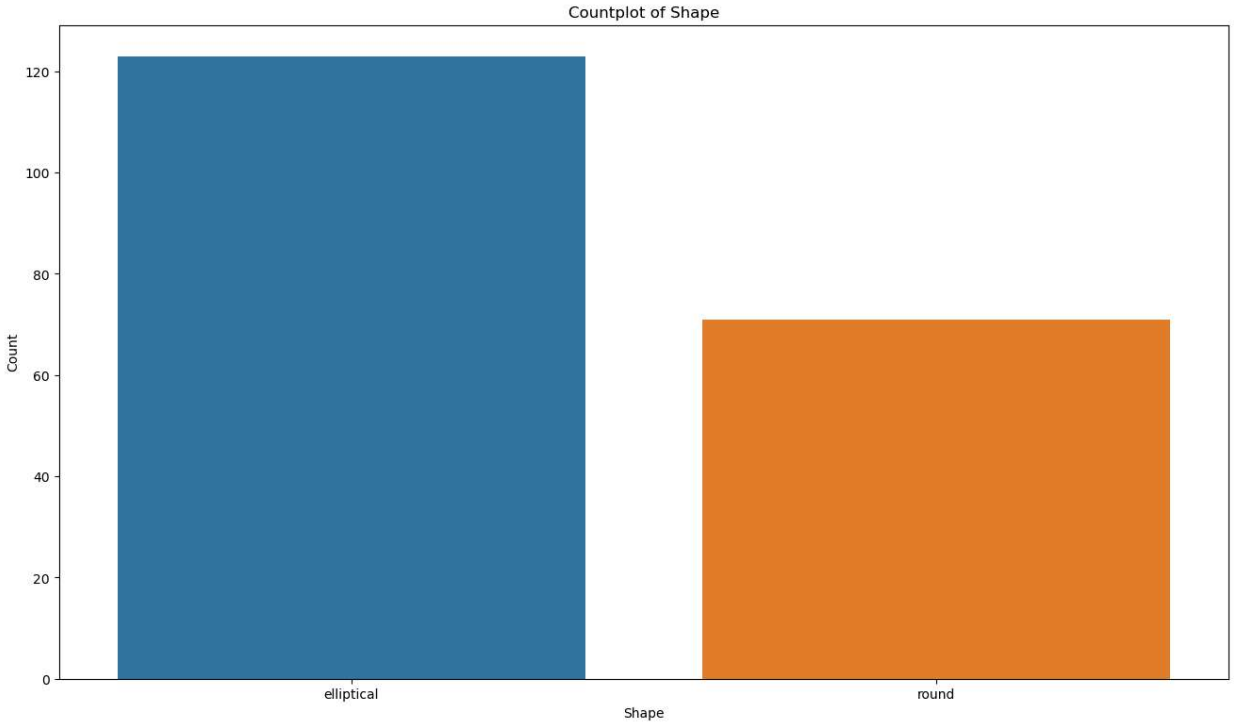
In [40]:
```
#Pie count
y.value_counts().plot.pie(autopct='%.2f%%',shadow=True)
plt.title('Pie Chart')
plt.ylabel('')
plt.show()
```



In [41]:
```
#05.Explore shape and soil type through countplot ,pie chart
```

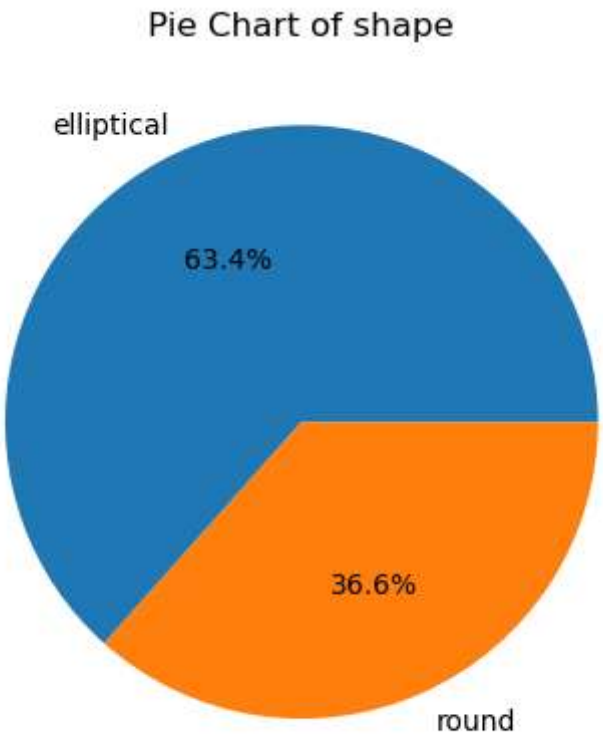
In [55]:
```
#countplot for shape
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(16, 9))
sns.countplot(x='SHAPE', data=df)
plt.title('Countplot of Shape')
plt.xlabel('Shape')
plt.ylabel('Count')
plt.show()
```



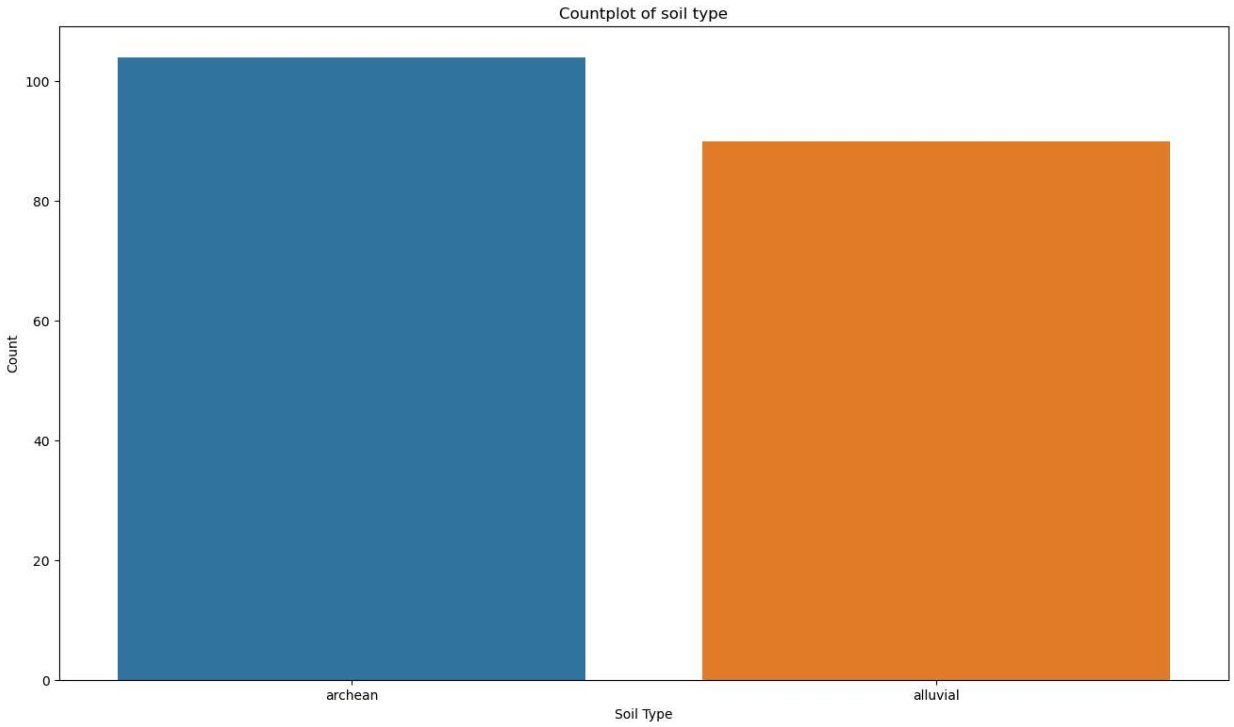
In [54]:
```
#Pie chart for shape
df['SHAPE'].value_counts().plot.pie(autopct='%1.1f%%')
plt.title('Pie Chart of shape')
plt.ylabel('')
```

Out[54]:
```
Text(0, 0.5, '')
```



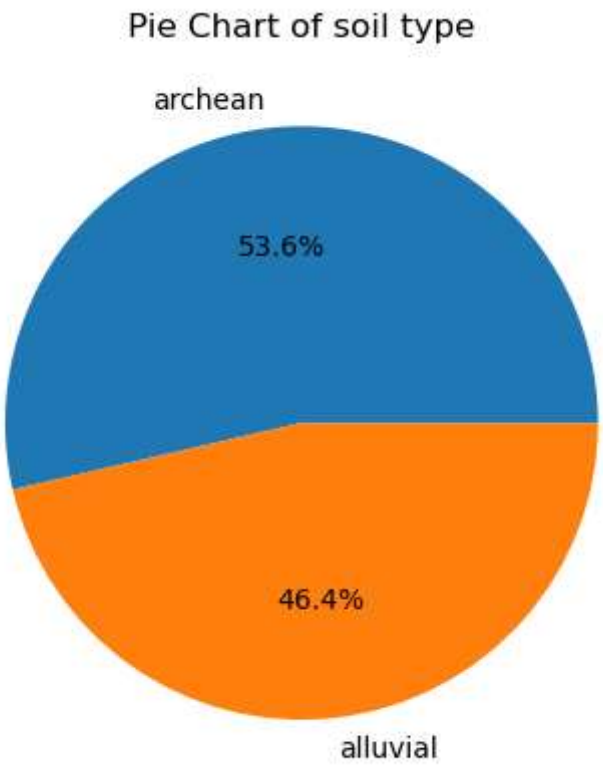
In [56]:
```
#countplot foe soil type
plt.figure(figsize=(16, 9))
sns.countplot(x='SOIL_TYPE', data=df)
```

```
plt.title('Countplot of soil type')
plt.xlabel('Soil Type')
plt.ylabel('Count')
plt.show()
```

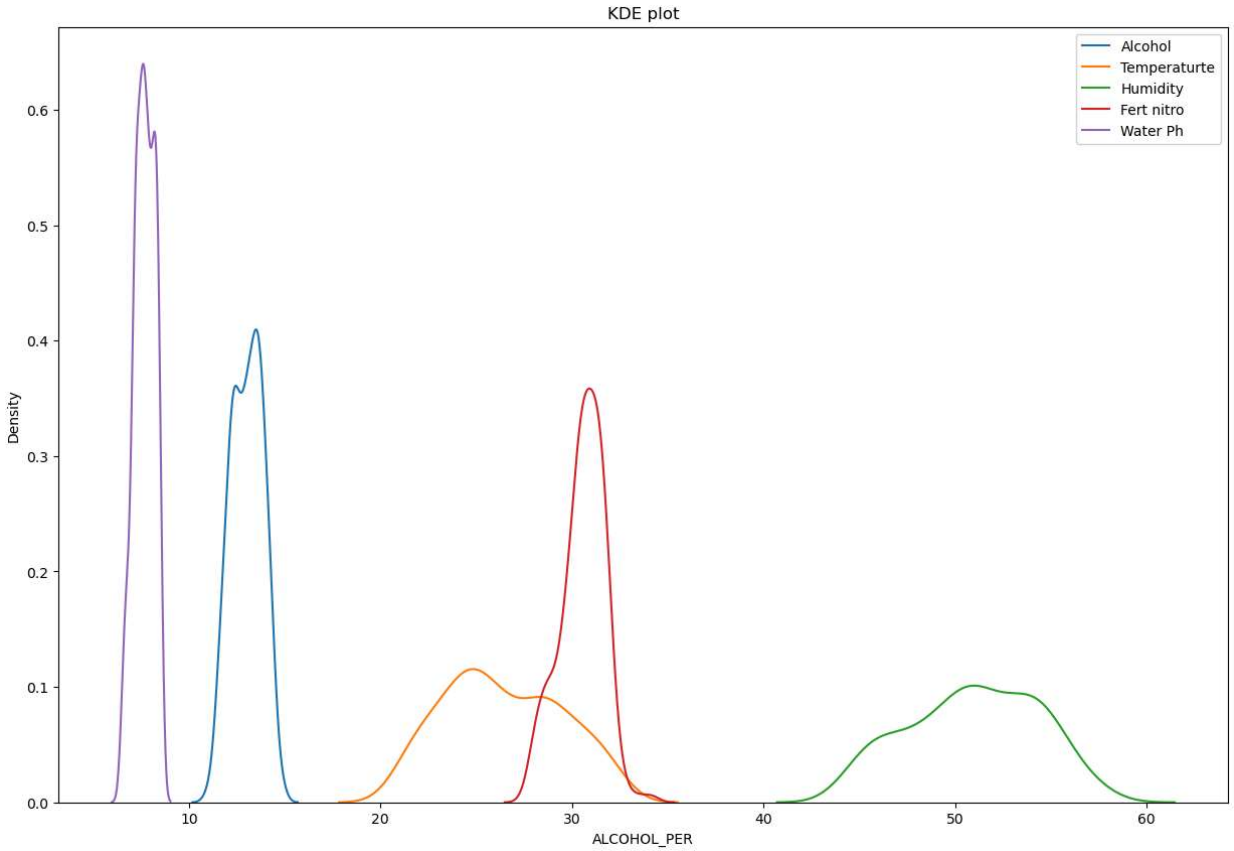


In [57]:
```
#Pie chart for soil type
df['SOIL_TYPE'].value_counts().plot.pie(autopct='%1.1f%%')
plt.title('Pie Chart of soil type')
plt.ylabel('')
```

Out[57]:
```
Text(0, 0.5, '')
```



In [70]:
```
#06.Explore alchohol_per, temperature, avg_humidity, fert_nitro_per, water_ph using ka
plt.figure(figsize=(15,10))
sns.kdeplot(x=df['ALCOHOL_PER'], data=df, label='Alcohol')
sns.kdeplot(x=df['TEMPERATURE'], data=df, label='Temperaturte')
sns.kdeplot(x=df['AVG_HUMIDITY'], data=df, label='Humidity')
sns.kdeplot(x=df['FERT_NITRO_PER'], data=df, label='Fert nitro')
sns.kdeplot(x=df['WATER_PH'], data=df, label='Water Ph')
plt.title('KDE plot')
plt.legend()
plt.show()
```



In [73]:
```
#07.Data Encoding
#for shape
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import accuracy_score
```

In [80]:
```
label_enc=LabelEncoder()
df['SHAPE_ENCODED']=label_enc.fit_transform(df['SHAPE'])
print(df[['SHAPE', 'SHAPE_ENCODED']])
```

```
     SHAPE  SHAPE_ENCODED
0        0              0
1        0              0
2        0              0
3        0              0
4        1              1
..     ...            ...
189      0              0
190      0              0
191      1              1
192      0              0
193      0              0

[194 rows x 2 columns]
```

In [81]:
```
#for soil type
label_enc=LabelEncoder()
df['SOIL_TYPE_ENCODED']=label_enc.fit_transform(df['SOIL_TYPE'])
print(df[['SOIL_TYPE','SOIL_TYPE_ENCODED']])
```

```
    SOIL_TYPE  SOIL_TYPE_ENCODED
0     archean                 1
1     archean                 1
2     archean                 1
3    alluvial                 0
4     archean                 1
..        ...               ...
189  alluvial                 0
190   archean                 1
191   archean                 1
192  alluvial                 0
193   archean                 1

[194 rows x 2 columns]
```

In [82]:
```
#08.Apply feature scaling
scaler=StandardScaler()
x_scaled=scaler.fit_transform(x)
```

In [83]:
```
x_scaled
```

Out[83]:
```
array([[ 0.2551645 ],
       [-0.30162623],
       [ 0.27265165],
       [ 1.2092632 ],
       [-0.32820669],
       [ 0.74550408],
       [ 0.08309099],
       [-0.00574372],
       [ 0.06070744],
       [ 0.95325138],
       [ 0.30342903],
       [-0.02742778],
       [ 0.23767736],
       [ 0.14884266],
       [ 1.07636089],
       [ 0.98822567],
       [ 0.50208301],
       [ 0.67905293],
       [ 1.60657116],
       [ 0.28174497],
       [ 0.23767736],
       [-0.29253291],
       [ 1.14281204],
       [ 1.48276217],
       [ 0.01663983],
       [ 0.25936142],
       [-0.24846531],
       [-0.55763805],
       [-0.50028021],
       [-0.68144704],
       [-0.65486658],
       [-0.11556299],
       [-0.49118689],
       [-0.24846531],
       [-0.16033009],
       [ 0.28174497],
       [ 0.81125575],
       [-0.5401509 ],
       [ 0.14884266],
       [-0.38066813],
       [ 0.01663983],
       [-0.2043977 ],
       [-0.35898407],
       [-0.60170566],
       [ 0.01663983],
       [ 0.47130563],
       [-0.34569384],
       [ 0.16213289],
       [-0.31002006],
       [-0.00994063],
       [ 0.07819459],
       [-0.07149539],
       [ 0.4580154 ],
       [ 0.50208301],
       [ 1.69470638],
       [ 0.94415806],
       [ 0.23767736],
       [ 0.87770691],
       [ 0.5468501 ],
       [ 0.34749664],
```

```
[ 0.52446656],
[ 0.5818244 ],
[ 0.41394779],
[ 0.76718814],
[-0.42473574],
[ 0.14884266],
[ 0.28174497],
[-0.02742778],
[ 0.1712262 ],
[ 0.90009046],
[-0.5352545 ],
[-0.02742778],
[ 0.28174497],
[-0.06310156],
[-0.2043977 ],
[ 0.23767736],
[-0.31491646],
[-0.29253291],
[ 1.39043003],
[-0.46950283],
[-0.88849486],
[-0.7430018 ],
[-0.58421851],
[-0.75629203],
[-0.81784678],
[-0.55763805],
[-0.07149539],
[ 1.14281204],
[ 1.48276217],
[ 1.91574391],
[ 1.56250355],
[ 2.53408939],
[ 0.90009046],
[ 2.4102804 ],
[ 1.1204285 ],
[ 1.25333081],
[ 1.74367039],
[ 1.07636089],
[ 3.50567524],
[ 2.9537809 ],
[ 1.14281204],
[ 0.36078687],
[ 0.22858404],
[ 0.09638122],
[ 1.99548529],
[ 0.68814624],
[ 2.48092847],
[ 2.29556472],
[ 1.58908401],
[ 1.51843594],
[ 0.19360975],
[ 2.13678143],
[ 2.04794673],
[ 1.16519559],
[ 0.98822567],
[ 2.26898426],
[ 1.8716763 ],
[ 1.82760869],
[-1.37463752],
[-0.79196581],
```

```
[ 0.30342903],
[-0.55763805],
[-0.27014937],
[-0.93326195],
[-0.2043977 ],
[ 0.10477505],
[-0.16872392],
[-0.83603342],
[-0.97732956],
[-0.88849486],
[-0.7430018 ],
[-0.58421851],
[-0.75629203],
[-0.81784678],
[-0.55763805],
[-0.2043977 ],
[-1.06546477],
[-0.73390848],
[-1.10043907],
[-1.13191593],
[-0.51357044],
[-1.26411875],
[-0.11556299],
[-0.88849486],
[-1.07875501],
[-1.15359999],
[-1.08784832],
[-0.99901362],
[-1.46766914],
[-1.17598354],
[-0.64577327],
[-0.88849486],
[-1.2865023 ],
[-0.80035964],
[-1.08784832],
[-1.13191593],
[-0.95494601],
[-0.24846531],
[-1.22005115],
[-0.77867558],
[-1.15359999],
[-0.99901362],
[-1.326373  ],
[-0.93745886],
[-1.04308123],
[-0.73390848],
[-0.77867558],
[-1.04308123],
[-1.06546477],
[-0.95494601],
[-1.35225397],
[-0.55763805],
[-0.87520462],
[-0.95494601],
[-1.39702107],
[-1.37463752],
[-1.326373  ],
[-0.73390848],
[-1.67052003],
[-0.80035964],
```

```
       [ 0.41394779],
       [-1.31727968],
       [ 0.74550408],
       [ 0.08309099],
       [-0.00574372],
       [ 0.06070744],
       [ 0.95325138],
       [-1.08784832],
       [-0.99901362],
       [-1.01720025],
       [-0.49538381],
       [-0.9108784 ],
       [-1.29979253],
       [-1.08784832]])
```

In [88]:
```
#09. Perform the cross validation
from sklearn.model_selection import train_test_split, cross_val_score
x_train,x_test,y_train,y_test=train_test_split(x_scaled,y,test_size=0.2,random_state=4
```

In [91]:
```
x_train
```

Out[91]:
```
array([[ 0.74550408],
       [-0.2043977 ],
       [-0.27014937],
       [-1.04308123],
       [-0.55763805],
       [ 0.23767736],
       [ 1.25333081],
       [ 0.28174497],
       [-1.01720025],
       [-0.35898407],
       [ 0.14884266],
       [ 1.99548529],
       [-0.77867558],
       [ 1.39043003],
       [-0.31491646],
       [-0.68144704],
       [-1.06546477],
       [ 0.52446656],
       [-0.07149539],
       [-0.73390848],
       [ 0.14884266],
       [-0.11556299],
       [-0.95494601],
       [ 0.23767736],
       [ 1.51843594],
       [-0.2043977 ],
       [-0.75629203],
       [ 1.14281204],
       [-0.24846531],
       [-0.97732956],
       [ 0.95325138],
       [ 3.50567524],
       [ 0.27265165],
       [-0.34569384],
       [-0.73390848],
       [ 1.56250355],
       [-1.15359999],
       [ 0.81125575],
       [ 0.06070744],
       [ 1.14281204],
       [-0.16872392],
       [-0.99901362],
       [-0.24846531],
       [-0.02742778],
       [-0.2043977 ],
       [-1.08784832],
       [ 0.08309099],
       [-0.55763805],
       [-0.99901362],
       [-0.32820669],
       [-0.49118689],
       [-0.88849486],
       [-0.51357044],
       [ 1.07636089],
       [-1.13191593],
       [ 0.30342903],
       [ 0.41394779],
       [-1.31727968],
       [-1.26411875],
       [ 0.2551645 ],
```

```
       [-1.39702107],
       [-1.46766914],
       [-0.5352545 ],
       [ 0.98822567],
       [-0.42473574],
       [ 0.01663983],
       [ 0.10477505],
       [-0.50028021],
       [ 0.01663983],
       [ 0.74550408],
       [-0.64577327],
       [-1.29979253],
       [ 0.25936142],
       [ 1.48276217],
       [-1.08784832],
       [-0.11556299],
       [-0.7430018 ],
       [ 1.58908401],
       [-0.46950283],
       [-0.81784678],
       [-0.38066813],
       [ 0.5818244 ],
       [ 0.30342903],
       [-0.07149539],
       [-0.29253291],
       [-0.99901362],
       [-0.93745886],
       [ 0.16213289],
       [ 1.1204285 ],
       [-0.73390848],
       [-0.60170566],
       [-0.95494601],
       [ 1.2092632 ],
       [ 0.68814624],
       [ 0.50208301],
       [-0.81784678],
       [-1.37463752],
       [-1.2865023 ],
       [ 0.41394779],
       [-0.00994063],
       [-0.88849486],
       [-0.16033009],
       [-0.00574372],
       [ 0.19360975],
       [ 2.53408939],
       [-0.75629203],
       [-0.55763805],
       [-1.326373  ],
       [ 1.91574391],
       [ 0.06070744],
       [ 0.14884266],
       [ 0.34749664],
       [-0.77867558],
       [-0.58421851],
       [ 0.67905293],
       [ 0.28174497],
       [-1.35225397],
       [-0.55763805],
       [-1.326373  ],
       [-1.67052003],
```

```
       [ 0.76718814],
       [ 1.69470638],
       [ 2.29556472],
       [ 0.07819459],
       [-0.49538381],
       [ 0.5468501 ],
       [-0.31002006],
       [ 1.48276217],
       [-0.29253291],
       [ 0.87770691],
       [-1.15359999],
       [-1.08784832],
       [-0.88849486],
       [-0.5401509 ],
       [-0.24846531],
       [ 0.08309099],
       [-0.30162623],
       [ 0.4580154 ],
       [-1.17598354],
       [-0.7430018 ],
       [-0.88849486],
       [ 0.09638122],
       [ 2.9537809 ],
       [ 1.8716763 ],
       [ 1.14281204],
       [-0.2043977 ],
       [-0.55763805],
       [-1.08784832],
       [ 0.23767736],
       [-0.02742778],
       [ 2.48092847],
       [ 1.07636089],
       [ 0.90009046],
       [-0.80035964],
       [ 0.22858404]])
```

In [92]:
```
x_test
```

Out[92]:
```
array([[-1.10043907],
       [ 0.50208301],
       [-1.13191593],
       [ 1.74367039],
       [ 0.1712262 ],
       [-0.80035964],
       [ 0.94415806],
       [ 0.98822567],
       [ 2.04794673],
       [ 2.13678143],
       [-0.00574372],
       [ 1.60657116],
       [-0.58421851],
       [ 0.95325138],
       [-1.04308123],
       [ 1.82760869],
       [ 0.90009046],
       [ 1.16519559],
       [-0.9108784 ],
       [-0.79196581],
       [-0.93326195],
       [-1.07875501],
       [ 0.28174497],
       [ 0.47130563],
       [-1.22005115],
       [ 2.26898426],
       [-0.02742778],
       [ 2.4102804 ],
       [-0.65486658],
       [ 0.36078687],
       [-1.37463752],
       [ 0.23767736],
       [ 0.01663983],
       [-0.87520462],
       [-0.83603342],
       [-0.95494601],
       [ 0.28174497],
       [-1.06546477],
       [-0.06310156]])
```

In [93]:
```
y_train
```

Out[93]:
```
5      elliptical
135    elliptical
122    elliptical
167    elliptical
85          round
          ...
106         round
14     elliptical
92          round
179         round
102         round
Name: SHAPE, Length: 155, dtype: object
```

In [94]:
```
y_test
```

Out[94]:
```
138    elliptical
16     elliptical
155    elliptical
96          round
68          round
153         round
55     elliptical
15     elliptical
112    elliptical
111         round
184    elliptical
18          round
82     elliptical
9      elliptical
164    elliptical
117         round
69     elliptical
113         round
191         round
119         round
123         round
144         round
66     elliptical
45     elliptical
158         round
115    elliptical
67     elliptical
93          round
30          round
101         round
118         round
75     elliptical
24     elliptical
172    elliptical
127    elliptical
169         round
19          round
168    elliptical
73     elliptical
Name: SHAPE, dtype: object
```

In [104…
```
#10.Implement Decision Tree algo
dt_model=DecisionTreeClassifier(random_state=42)
dt_model.fit(x_train,y_train)
```

Out[104]:
```
DecisionTreeClassifier(random_state=42)
```

In [106…
```
y_pred=dt_model.predict(x_test)
accuracy=accuracy_score(y_test,y_pred)
```

In [107…
```
accuracy
```

Out[107]:
```
0.6153846153846154
```

In [ ]: