In [1]:
```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
```

In [45]:
```
df=pd.read_csv('Training_Data.csv')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   area               2000 non-null   int64  
 1   major_axis_length  2000 non-null   float64
 2   minor_axis_length  2000 non-null   float64
 3   eccentricity       2000 non-null   float64
 4   equiv_diameter     2000 non-null   float64
 5   solidity           2000 non-null   float64
 6   extent             2000 non-null   float64
 7   perimeter          2000 non-null   float64
 8   aspect_ratio       2000 non-null   float64
 9   compactness        2000 non-null   float64
 10  roundness          2000 non-null   float64
 11  category           2000 non-null   object 
dtypes: float64(10), int64(1), object(1)
memory usage: 187.6+ KB
```

In [46]:
```
df.columns
```

Out[46]:
```
Index(['area', 'major_axis_length', 'minor_axis_length', 'eccentricity',
       'equiv_diameter', 'solidity', 'extent', 'perimeter', 'aspect_ratio',
       'compactness', 'roundness', 'category'],
      dtype='object')
```

In [47]:
```
df.describe()
```

Out[47]:

	area	major_axis_length	minor_axis_length	eccentricity	equiv_diameter	solidity	extent	perimeter	aspect_ratio
count	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000
mean	756.563500	48.263838	20.004575	0.852816	30.303021	0.962658	0.642215	109.856107	2.386041
std	307.328187	17.575851	2.016167	0.125772	6.710886	0.010875	0.121566	32.217054	0.793757
min	55.000000	10.000590	7.258053	0.176386	8.368284	0.844311	0.355283	24.174000	1.015929
25%	490.750000	32.385686	18.983436	0.792793	24.996843	0.957130	0.544887	81.602000	1.640713
50%	760.000000	47.976010	20.420683	0.904524	31.107267	0.962963	0.647763	111.304500	2.345094
75%	1042.250000	65.462575	21.393419	0.947681	36.428476	0.969340	0.741110	140.554000	3.132633
max	1354.000000	77.179776	24.396405	0.972206	41.520674	0.993094	0.912500	166.226000	4.271183

In [48]:
```
X=df[['area', 'major_axis_length', 'minor_axis_length', 'perimeter',
       'eccentricity', 'solidity', 'extent', 'equiv_diameter', 'aspect_ratio',
       'compactness', 'roundness']]
```

In [49]:
```
unique_categories = df['category'].unique()
print(unique_categories)
```

```
['Small Broke C1' 'Small Broke' 'Big Broke' 'Head rice' 'Whole Rice']
```

In [50]:
```
y=df[['category']]
y.describe()
```

Out[50]:

	category
count	2000
unique	5
top	Small Broke C1
freq	400

In [51]:
```
import pandas as pd

# Assuming 'df' is your DataFrame with the 'Category' column
df['category'] = df['category'].replace({
    'Small Broke C1': '1',
    'Small Broke': '2',
    'Big Broke': '3',
    'Head rice': '4',
    'Whole Rice': '5'
})
```

```
# Verify the updated 'Category' column
```

In [52]:
```
print(df['category'])
```

```
0       1
1       1
2       1
3       1
4       1
       ..
1995    5
1996    5
1997    5
1998    5
1999    5
Name: category, Length: 2000, dtype: object
```

In [53]:
```
y=df[['category']]
```

In [54]:
```
import pandas as pd
import numpy as np

np.random.seed(42)

X = pd.DataFrame(X)  # Convert X to a DataFrame

X_shadow = X.apply(np.random.permutation)
X_shadow.columns = ['shadow_' + str(feat) for feat in X.columns]
X_boruta = pd.concat([X, X_shadow], axis=1)
```

In [55]:
```
X_shadow.columns
```

Out[55]:
```
Index(['shadow_area', 'shadow_major_axis_length', 'shadow_minor_axis_length',
       'shadow_perimeter', 'shadow_eccentricity', 'shadow_solidity',
       'shadow_extent', 'shadow_equiv_diameter', 'shadow_aspect_ratio',
       'shadow_compactness', 'shadow_roundness'],
      dtype='object')
```

In [56]:
```
X_boruta
```

Out[56]:

	area	major_axis_length	minor_axis_length	perimeter	eccentricity	solidity	extent	equiv_diameter	aspect_ratio	compactness	...
0	447	28.993104	20.125638	76.234	0.719828	0.971739	0.840226	23.856615	1.440605	0.822838	...
1	440	27.778925	20.823307	75.582	0.661882	0.977778	0.679012	23.669081	1.334030	0.852052	...
2	409	24.636357	21.672122	71.752	0.475565	0.962353	0.711304	22.820056	1.136776	0.926276	...
3	374	27.775012	17.482554	70.561	0.777053	0.968912	0.785714	21.821815	1.588727	0.785664	...
4	428	28.919643	19.338227	74.772	0.743543	0.974943	0.636905	23.344090	1.495465	0.807205	...
...	...	...	...	...	...	...	...	...	...	...	...
1995	1241	70.046907	22.994696	150.715	0.944582	0.966511	0.707123	39.750349	3.046220	0.567482	...
1996	1154	70.467498	21.360438	148.490	0.952951	0.957676	0.832612	38.331690	3.298973	0.543963	...
1997	1221	70.149500	22.623034	151.642	0.946570	0.961417	0.517373	39.428739	3.100800	0.562067	...
1998	1128	68.653255	21.410320	147.498	0.950128	0.957555	0.629464	37.897417	3.206550	0.552012	...
1999	1199	67.612435	23.111918	147.523	0.939762	0.958433	0.623829	39.071911	2.925436	0.577881	...

2000 rows × 22 columns

In [57]:
```
from sklearn.ensemble import RandomForestRegressor

forest = RandomForestRegressor(max_depth=10, random_state=42)
forest.fit(X_boruta, y)
# Store feature importances
feat_imp_X = forest.feature_importances_[:len(X.columns)]
feat_imp_shadow = forest.feature_importances_[len(X.columns):]
# Compute hits
hits= feat_imp_X > feat_imp_shadow.max()
```

```
C:\Users\LENOVO\AppData\Local\Temp\ipykernel_4416\619982654.py:4: DataConversionWarning: A column-vector y was
passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
  forest.fit(X_boruta, y)
```

In [58]:
```
import numpy as np

arr = np.array([feat_imp_X])

np.set_printoptions(formatter={'float': lambda x: format(x, '10.10f')})
print(arr)
```

```
[[0.2048170656 0.2895658364 0.0000924527 0.2846858513 0.0002184748
  0.0003550452 0.0003123676 0.2126686310 0.0002247837 0.0002052591
  0.0002689537]]
```

In [59]:
```
hits
```

Out[59]:
```
array([ True,  True, False,  True, False, False, False,  True, False,
       False, False])
```

In [60]:
```
feat_imp_shadow
```

Out[60]:
```
array([0.0007447264, 0.0006019374, 0.0006603431, 0.0005292658,
       0.0005652549, 0.0004168196, 0.0007439953, 0.0006495917,
       0.0005652226, 0.0005156588, 0.0005924633])
```

In [61]:
```
fs = pd.Series(data=hits, index=X.columns)
fs
```

Out[61]:
```
area                 True
major_axis_length    True
minor_axis_length   False
perimeter            True
eccentricity        False
solidity            False
extent              False
equiv_diameter       True
aspect_ratio        False
compactness         False
roundness           False
dtype: bool
```

In [62]:
```
fss = pd.Series(data=feat_imp_shadow, index=X.columns)#[hits] #.sort_values(ascending=True)
fss
```

Out[62]:
```
area                0.000745
major_axis_length   0.000602
minor_axis_length   0.000660
perimeter           0.000529
eccentricity        0.000565
solidity            0.000417
extent              0.000744
equiv_diameter      0.000650
aspect_ratio        0.000565
compactness         0.000516
roundness           0.000592
dtype: float64
```

In [63]:
```
#fs = pd.Series(feat_imp_X, index = X.columns).sort_values (ascending= True)
fs = pd.Series(data=feat_imp_X, index=X.columns)[hits].sort_values(ascending=True)

fs
```

Out[63]:
```
area                0.204817
equiv_diameter      0.212669
perimeter           0.284686
major_axis_length   0.289566
dtype: float64
```

In [ ]:
```

```