Training ML and selecting the best ML

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

In [2]:
```
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 [3]:
```
df.columns
```

Out[3]:
```
Index(['area', 'major_axis_length', 'minor_axis_length', 'eccentricity',
       'equiv_diameter', 'solidity', 'extent', 'perimeter', 'aspect_ratio',
       'compactness', 'roundness', 'category'],
      dtype='object')
```

In [4]:
```
X=df[['area', 'major_axis_length', 'minor_axis_length', 'perimeter',
       'eccentricity', 'solidity', 'extent', 'equiv_diameter', 'aspect_ratio',
       'compactness', 'roundness']]
```

In [5]:
```
import pandas as pd


df['category'] = df['category'].replace({
    'Small Broke C1': '1',
    'Small Broke': '2',
    'Big Broke': '3',
    'Head rice': '4',
    'Whole Rice': '5'
})
```

In [6]:
```
y=df[['category']]
```

In [7]:
```
y
```

Out[7]:

	category
0	1
1	1
2	1
3	1
4	1
...	...
1995	5
1996	5
1997	5
1998	5
1999	5

2000 rows × 1 columns

In [ ]:
```

```

In [8]:
```
X1=df[['area', 'major_axis_length',  'perimeter','equiv_diameter' ]] # According to the result of Boruta's feat
```

In [9]:
```
X1
```

Out[9]:

	area	major_axis_length	perimeter	equiv_diameter
0	447	28.993104	76.234	23.856615
1	440	27.778925	75.582	23.669081
2	409	24.636357	71.752	22.820056
3	374	27.775012	70.561	21.821815
4	428	28.919643	74.772	23.344090
...	...	...	...	...
1995	1241	70.046907	150.715	39.750349
1996	1154	70.467498	148.490	38.331690
1997	1221	70.149500	151.642	39.428739
1998	1128	68.653255	147.498	37.897417
1999	1199	67.612435	147.523	39.071911

2000 rows × 4 columns

In [10]:
```
from sklearn.model_selection import train_test_split

#Splitting training and validating data
X1_train, X1_test, y_train, y_test = train_test_split (X1,y, test_size= 0.3, random_state=7, shuffle=True)
```

In [11]:
```
import sklearn
from sklearn.model_selection import train_test_split

# classifiers algorithm
from sklearn.neighbors import KNeighborsClassifier                    # 1. K-Neighbors Classifier
from sklearn.linear_model import LogisticRegression                   # 2. Logistic Regression Classifier
from sklearn.tree import DecisionTreeClassifier                           # 3. Decision Tree Classifier
from sklearn.ensemble import GradientBoostingClassifier             # 4. Gradient Boosting Classifier
from sklearn.ensemble import RandomForestClassifier                 # 5. Random Forest Classifier
from sklearn.ensemble import BaggingClassifier                          # 6. Bagging Classifier
from sklearn.ensemble import AdaBoostClassifier                       # 7. Ada Boost Classifier
from sklearn.naive_bayes import GaussianNB                            # 8. Gaussian NB Classifier
from sklearn.neural_network import MLPClassifier                      # 9. Multilayer Perceptron
from sklearn.svm import SVC                                                   # 10. Support Vector Classifier
from sklearn.gaussian_process import GaussianProcessClassifier   # 11. Gaussian Process Classifier


from sklearn import metrics
import time
```

In [12]:
```
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

algo = [
    [KNeighborsClassifier(n_neighbors=4), 'KNeighborsClassifier'],
    [LogisticRegression(solver='liblinear'), 'LogisticRegression'],
    [DecisionTreeClassifier(min_samples_split=3), 'DecisionTreeClassifier'],
    [GradientBoostingClassifier(), 'GradientBoostingClassifier'],
    [RandomForestClassifier(), 'RandomForestClassifier'],
    [BaggingClassifier(), 'BaggingClassifier'],
    [AdaBoostClassifier(n_estimators=5), 'AdaBoostClassifier'],
    [GaussianNB(), 'GaussianNB'],
    [MLPClassifier(), 'MLPClassifier'],
    [SVC(kernel='linear'), 'SVC_linear'],
    [GaussianProcessClassifier(), 'GaussianProcessClassifier']
]

top_models = []  # List to store the top models
top_scores = []  # List to store the corresponding scores

for a in algo:
    model = a[0]
    start_time = time.time()
    model.fit(X1_train, y_train)
    end_time = time.time()
    total_time = end_time - start_time
    y_pred = model.predict(X1_test)

    # Calculate evaluation metrics
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred, average='weighted')
    recall = recall_score(y_test, y_pred, average='weighted')
    f1 = f1_score(y_test, y_pred, average='weighted')

    print(f"Model: {a[1]}")
    print(f"Total Computation Time: {total_time:.4f} seconds")
    report = classification_report(y_test, y_pred, digits=4)
```

```
    print(report)

    # Calculate a combined score
    combined_score = (accuracy + precision + recall + f1) / 4

    # Add the model and its score to the top_models and top_scores lists
    top_models.append(a[1])
    top_scores.append(combined_score)

# Sort the models based on scores in descending order
sorted_models = [x for _, x in sorted(zip(top_scores, top_models), reverse=True)]
```

```
C:\Users\LENOVO\anaconda3\envs\python3\Lib\site-packages\sklearn\neighbors\_classification.py:215: 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().
  return self._fit(X, y)
C:\Users\LENOVO\anaconda3\envs\python3\Lib\site-packages\sklearn\utils\validation.py:1143: 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().
  y = column_or_1d(y, warn=True)
Model: KNeighborsClassifier
Total Computation Time: 0.0140 seconds
              precision    recall  f1-score   support

           1     0.9781    1.0000    0.9889       134
           2     0.9440    0.9752    0.9593       121
           3     0.9619    0.9099    0.9352       111
           4     0.9569    0.9487    0.9528       117
           5     0.9829    0.9829    0.9829       117

    accuracy                         0.9650       600
   macro avg     0.9648    0.9633    0.9638       600
weighted avg     0.9650    0.9650    0.9648       600

Model: LogisticRegression
Total Computation Time: 0.0156 seconds
              precision    recall  f1-score   support

           1     0.9559    0.9701    0.9630       134
           2     0.7647    0.8595    0.8093       121
           3     0.7727    0.6126    0.6834       111
           4     0.8509    0.8291    0.8398       117
           5     0.9206    0.9915    0.9547       117

    accuracy                         0.8583       600
   macro avg     0.8530    0.8526    0.8501       600
weighted avg     0.8561    0.8583    0.8546       600

Model: DecisionTreeClassifier
Total Computation Time: 0.0156 seconds
              precision    recall  f1-score   support

           1     1.0000    1.0000    1.0000       134
           2     0.9758    1.0000    0.9878       121
           3     0.9266    0.9099    0.9182       111
           4     0.8824    0.8974    0.8898       117
           5     0.9649    0.9402    0.9524       117

    accuracy                         0.9517       600
   macro avg     0.9499    0.9495    0.9496       600
weighted avg     0.9518    0.9517    0.9516       600
```

```
C:\Users\LENOVO\anaconda3\envs\python3\Lib\site-packages\sklearn\ensemble\_gb.py:437: 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().
  y = column_or_1d(y, warn=True)
Model: GradientBoostingClassifier
Total Computation Time: 1.4566 seconds
              precision    recall  f1-score   support

           1     1.0000    1.0000    1.0000       134
           2     0.9917    0.9917    0.9917       121
           3     0.9626    0.9279    0.9450       111
           4     0.8871    0.9402    0.9129       117
           5     0.9649    0.9402    0.9524       117

    accuracy                         0.9617       600
   macro avg     0.9613    0.9600    0.9604       600
weighted avg     0.9626    0.9617    0.9619       600
```

```
C:\Users\LENOVO\AppData\Local\Temp\ipykernel_4204\4228527623.py:24: 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().
  model.fit(X1_train, y_train)
```

```
Model: RandomForestClassifier
Total Computation Time: 0.2774 seconds
              precision    recall  f1-score   support

           1     1.0000    1.0000    1.0000       134
           2     0.9758    1.0000    0.9878       121
           3     0.9806    0.9099    0.9439       111
           4     0.9174    0.9487    0.9328       117
           5     0.9661    0.9744    0.9702       117

    accuracy                         0.9683       600
   macro avg     0.9680    0.9666    0.9669       600
weighted avg     0.9688    0.9683    0.9682       600

Model: BaggingClassifier
Total Computation Time: 0.0469 seconds
              precision    recall  f1-score   support

           1     1.0000    1.0000    1.0000       134
           2     0.9758    1.0000    0.9878       121
           3     0.9619    0.9099    0.9352       111
           4     0.9016    0.9402    0.9205       117
           5     0.9739    0.9573    0.9655       117

    accuracy                         0.9633       600
   macro avg     0.9627    0.9615    0.9618       600
weighted avg     0.9638    0.9633    0.9633       600

Model: AdaBoostClassifier
Total Computation Time: 0.0313 seconds
              precision    recall  f1-score   support

           1     0.0000    0.0000    0.0000       134
           2     0.4627    0.9752    0.6277       121
           3     0.9722    0.9459    0.9589       111
           4     0.9658    0.9658    0.9658       117
           5     0.9667    0.9915    0.9789       117

    accuracy                         0.7533       600
   macro avg     0.6735    0.7757    0.7063       600
weighted avg     0.6500    0.7533    0.6832       600

Model: GaussianNB
Total Computation Time: 0.0000 seconds
              precision    recall  f1-score   support

           1     0.9851    0.9851    0.9851       134
           2     0.9440    0.9752    0.9593       121
           3     0.9697    0.8649    0.9143       111
           4     0.8934    0.9316    0.9121       117
           5     0.9500    0.9744    0.9620       117

    accuracy                         0.9483       600
   macro avg     0.9484    0.9462    0.9466       600
weighted avg     0.9492    0.9483    0.9481       600
```

```
C:\Users\LENOVO\anaconda3\envs\python3\Lib\site-packages\sklearn\ensemble\_bagging.py:802: DataConversionWarnin
g: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), fo
r example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\LENOVO\anaconda3\envs\python3\Lib\site-packages\sklearn\utils\validation.py:1143: DataConversionWarnin
g: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), fo
r example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\LENOVO\anaconda3\envs\python3\Lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetr
icWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_divisio
n` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\LENOVO\anaconda3\envs\python3\Lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetr
icWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\LENOVO\anaconda3\envs\python3\Lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetr
icWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\LENOVO\anaconda3\envs\python3\Lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetr
icWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\LENOVO\anaconda3\envs\python3\Lib\site-packages\sklearn\utils\validation.py:1143: DataConversionWarnin
g: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), fo
r example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\LENOVO\anaconda3\envs\python3\Lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:1098:
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().
  y = column_or_1d(y, warn=True)
```

```
Model: MLPClassifier
Total Computation Time: 0.4063 seconds
              precision    recall  f1-score   support

           1     0.8881    0.9478    0.9170       134
           2     0.8154    0.4380    0.5699       121
           3     0.5543    0.8739    0.6783       111
           4     0.8333    0.2991    0.4403       117
           5     0.6229    0.9316    0.7466       117

    accuracy                         0.7017       600
   macro avg     0.7428    0.6981    0.6704       600
weighted avg     0.7493    0.7017    0.6766       600

Model: SVC_linear
Total Computation Time: 0.0781 seconds
              precision    recall  f1-score   support

           1     0.9924    0.9701    0.9811       134
           2     0.9219    0.9752    0.9478       121
           3     0.9709    0.9009    0.9346       111
           4     0.9402    0.9402    0.9402       117
           5     0.9504    0.9829    0.9664       117

    accuracy                         0.9550       600
   macro avg     0.9551    0.9539    0.9540       600
weighted avg     0.9558    0.9550    0.9549       600

C:\Users\LENOVO\anaconda3\envs\python3\Lib\site-packages\sklearn\utils\validation.py:1143: DataConversionWarnin
g: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), fo
r example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\LENOVO\anaconda3\envs\python3\Lib\site-packages\sklearn\utils\validation.py:1143: DataConversionWarnin
g: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), fo
r example using ravel().
  y = column_or_1d(y, warn=True)
Model: GaussianProcessClassifier
Total Computation Time: 8.0387 seconds
              precision    recall  f1-score   support

           1     0.9925    0.9925    0.9925       134
           2     0.9835    0.9835    0.9835       121
           3     0.9533    0.9189    0.9358       111
           4     0.8926    0.9231    0.9076       117
           5     0.9487    0.9487    0.9487       117

    accuracy                         0.9550       600
   macro avg     0.9541    0.9533    0.9536       600
weighted avg     0.9554    0.9550    0.9551       600
```

In [13]:
```python
# Print the top 3 models
print("Top 3 Models:")
for i, model in enumerate(sorted_models[:3]):
    print(f"{i + 1}. {model}")
```

```
Top 3 Models:
1. RandomForestClassifier
2. KNeighborsClassifier
3. BaggingClassifier
```

In [ ]: