```python
# Importing libraries
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import plot_tree
from sklearn.metrics import classification_report,confusion_matrix

import pandas as pd

file_path = r"C:\PRAGYA\UST\COURSES\SEM 4\Machine Learning\
Assignments\HW9\ML_HW_Data_CancerGene.xlsx"

# Importing the data from different sheets of the Excel file
Cancer_Gene_train_X = pd.read_excel(file_path, sheet_name=0,
header=None)
Cancer_Gene_train_labels = pd.read_excel(file_path, sheet_name=1,
header=None)
Cancer_Gene_test_X = pd.read_excel(file_path, sheet_name=2,
header=None)
Cancer_Gene_test_labels = pd.read_excel(file_path, sheet_name=3,
header=None)

# Veryfing test_lables
Cancer_Gene_test_labels.head()
```

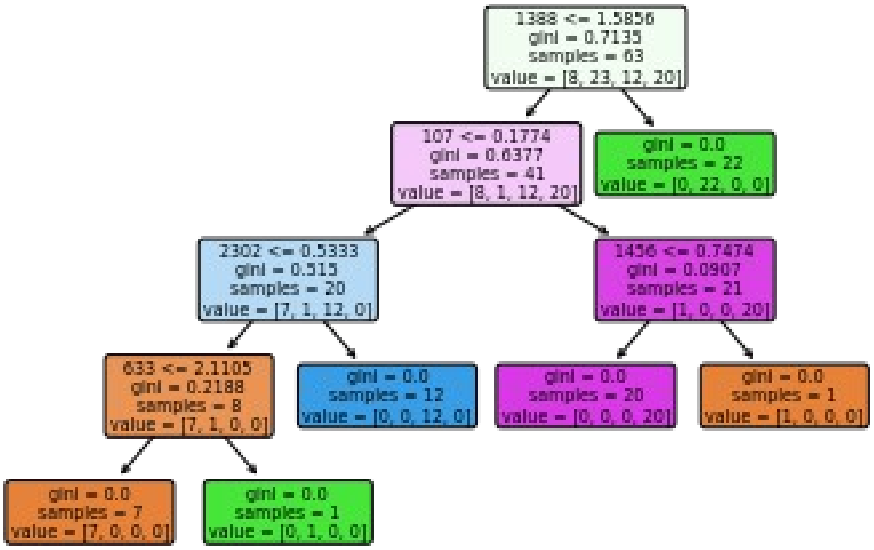
```
     0
0   NB
1   EW
2  NaN
3   RM
4  NaN
```

```python
# Python interprets the class 'NA' as if it were a NaN. So change the
class label from 'NA' to 'NAA'.
Cancer_Gene_test_labels.loc[Cancer_Gene_test_labels[0].isnull(), 0] =
'NAA'

# Training and fitting the DecisionTreeClassifier on the Training set
DT_Model = DecisionTreeClassifier(criterion = 'gini' )
DT_Model.fit(Cancer_Gene_train_X, Cancer_Gene_train_labels )
```

```
DecisionTreeClassifier()
```

```python
# Plotting the decision tree
DecTree = plot_tree(decision_tree=DT_Model, feature_names =
Cancer_Gene_train_X.columns, filled = True , precision = 4, rounded =
True)
```



```
# Predicition of Training Data
Predict_train = DT_Model.predict(Cancer_Gene_train_X)

# CFM for Training Data
confusion_matrix(Cancer_Gene_train_labels, Predict_train)

array([[ 8,  0,  0,  0],
       [ 0, 23,  0,  0],
       [ 0,  0, 12,  0],
       [ 0,  0,  0, 20]], dtype=int64)

# Classification Report for Training Data
print(classification_report(Cancer_Gene_train_labels, Predict_train))

              precision    recall  f1-score   support

          BL       1.00      1.00      1.00         8
          EW       1.00      1.00      1.00        23
          NB       1.00      1.00      1.00        12
          RM       1.00      1.00      1.00        20

    accuracy                           1.00        63
   macro avg       1.00      1.00      1.00        63
weighted avg       1.00      1.00      1.00        63


# Prediction of Test Data
Predict_test = DT_Model.predict(Cancer_Gene_test_X)

# CFM for Test Data
confusion_matrix(Cancer_Gene_test_labels, Predict_test)
```

```
array([[1, 1, 0, 1, 0],
       [0, 4, 0, 1, 1],
       [1, 0, 0, 1, 3],
       [0, 0, 0, 5, 1],
       [0, 1, 0, 0, 4]], dtype=int64)
```

```
# Classification Report for Test Data
print(classification_report(Cancer_Gene_test_labels, Predict_test))
```

```
              precision    recall  f1-score   support

          BL       0.50      0.33      0.40         3
          EW       0.67      0.67      0.67         6
         NAA       0.00      0.00      0.00         5
          NB       0.62      0.83      0.71         6
          RM       0.44      0.80      0.57         5

    accuracy                           0.56        25
   macro avg       0.45      0.53      0.47        25
weighted avg       0.46      0.56      0.49        25


C:\Users\pragy\anaconda3\lib\site-packages\sklearn\metrics\
_classification.py:1318: UndefinedMetricWarning: 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\pragy\anaconda3\lib\site-packages\sklearn\metrics\
_classification.py:1318: UndefinedMetricWarning: 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\pragy\anaconda3\lib\site-packages\sklearn\metrics\
_classification.py:1318: UndefinedMetricWarning: 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))
```