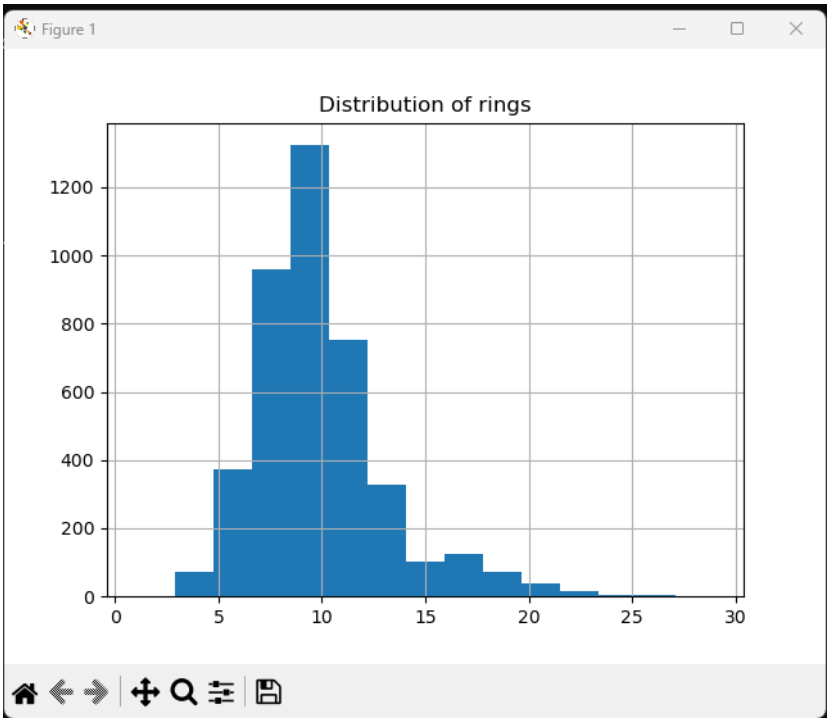
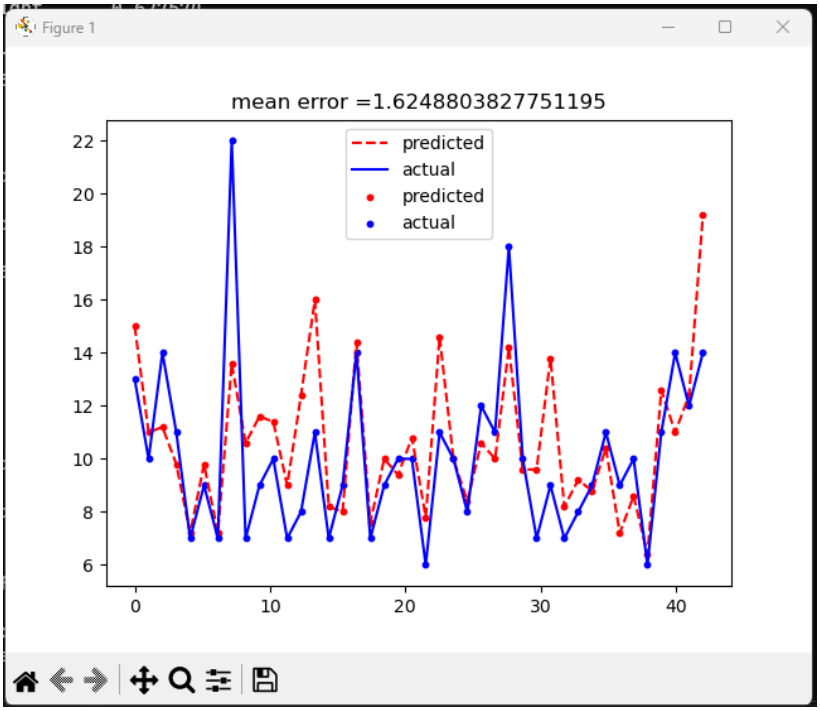
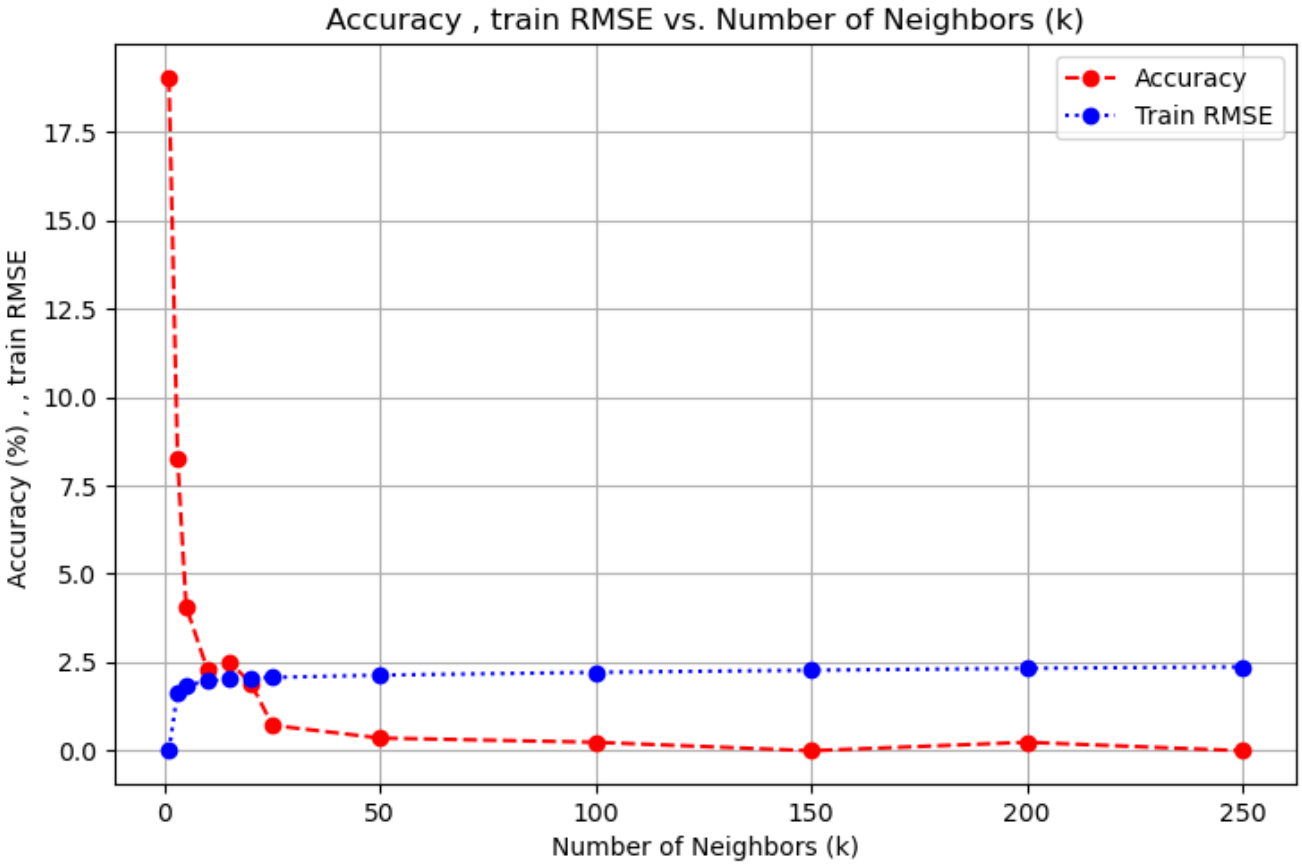
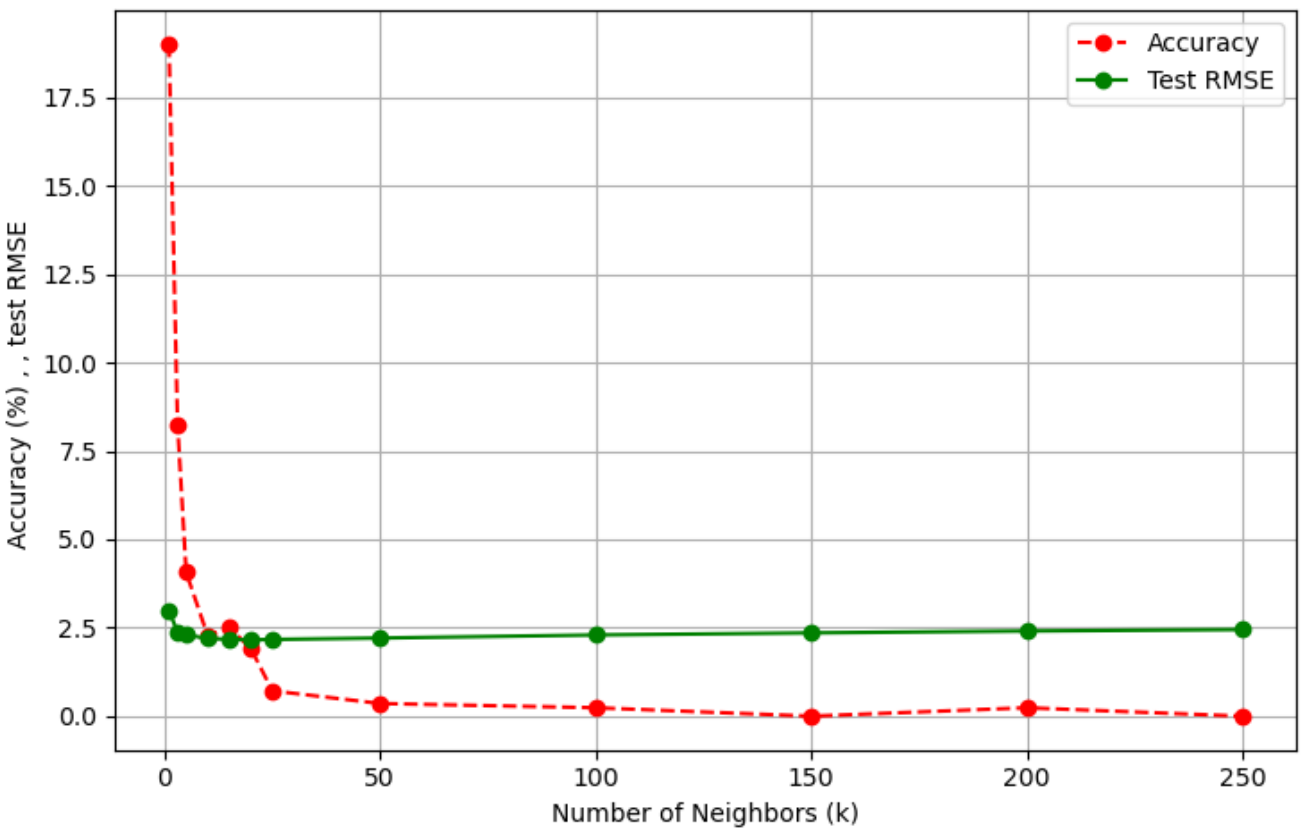
ASSIGNMENT 04

PROBLEM #01 (ABALONE) SOLUTION OUTPUT



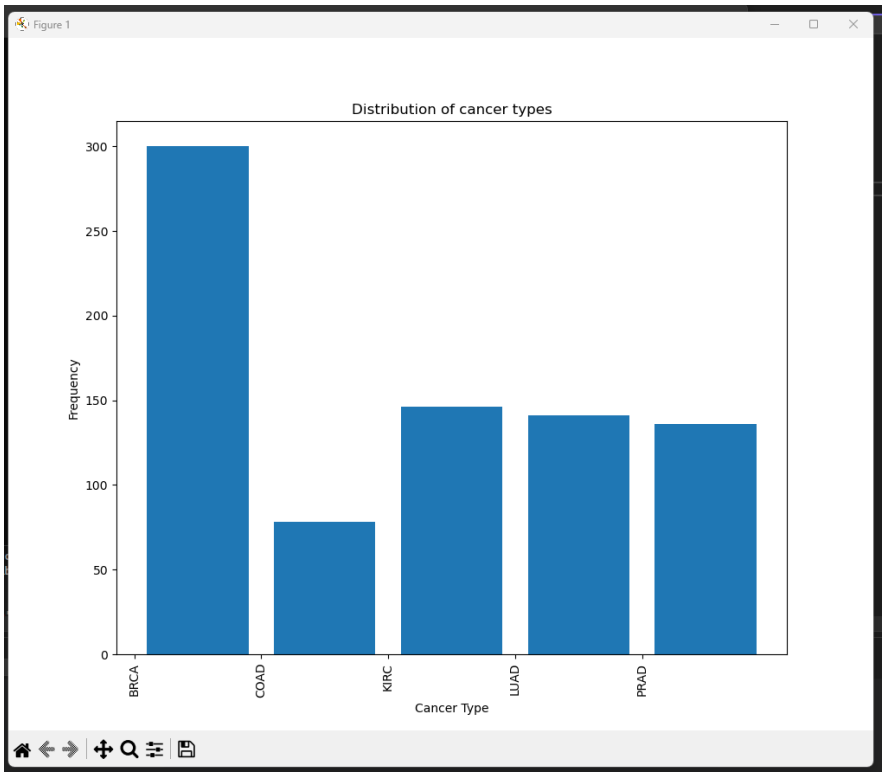


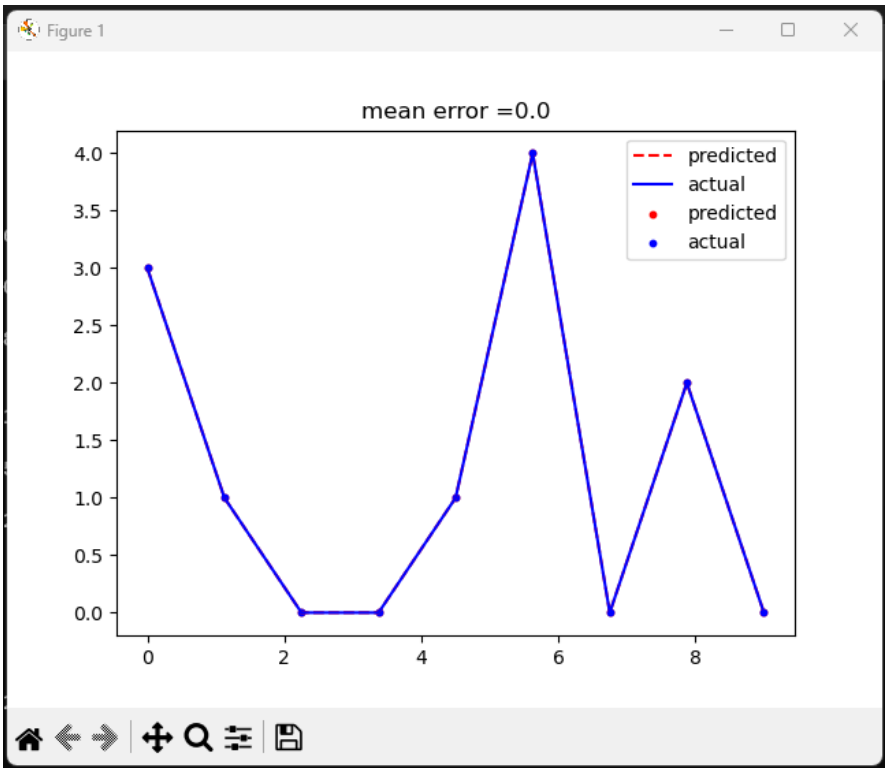

Accuracy , train RMSE vs. Number of Neighbors (k)
Accuracy
Train RMSE
Accuracy (%) , , train RMSE
Number of Neighbors (k)
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
0
50
100
150
200
250

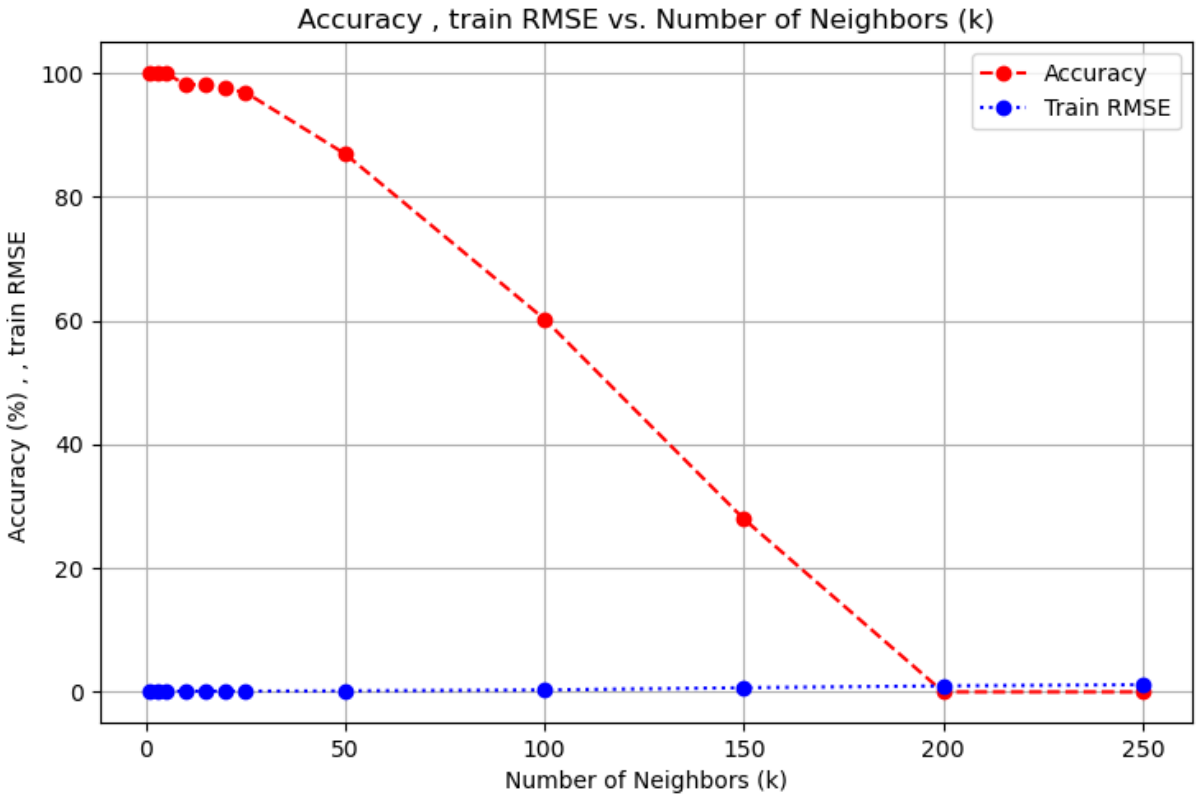

Accuracy , test RMSE vs. Number of Neighbors (k)
Accuracy
Test RMSE
Accuracy (%) , , test RMSE
Number of Neighbors (k)
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
0
50
100
150
200
250

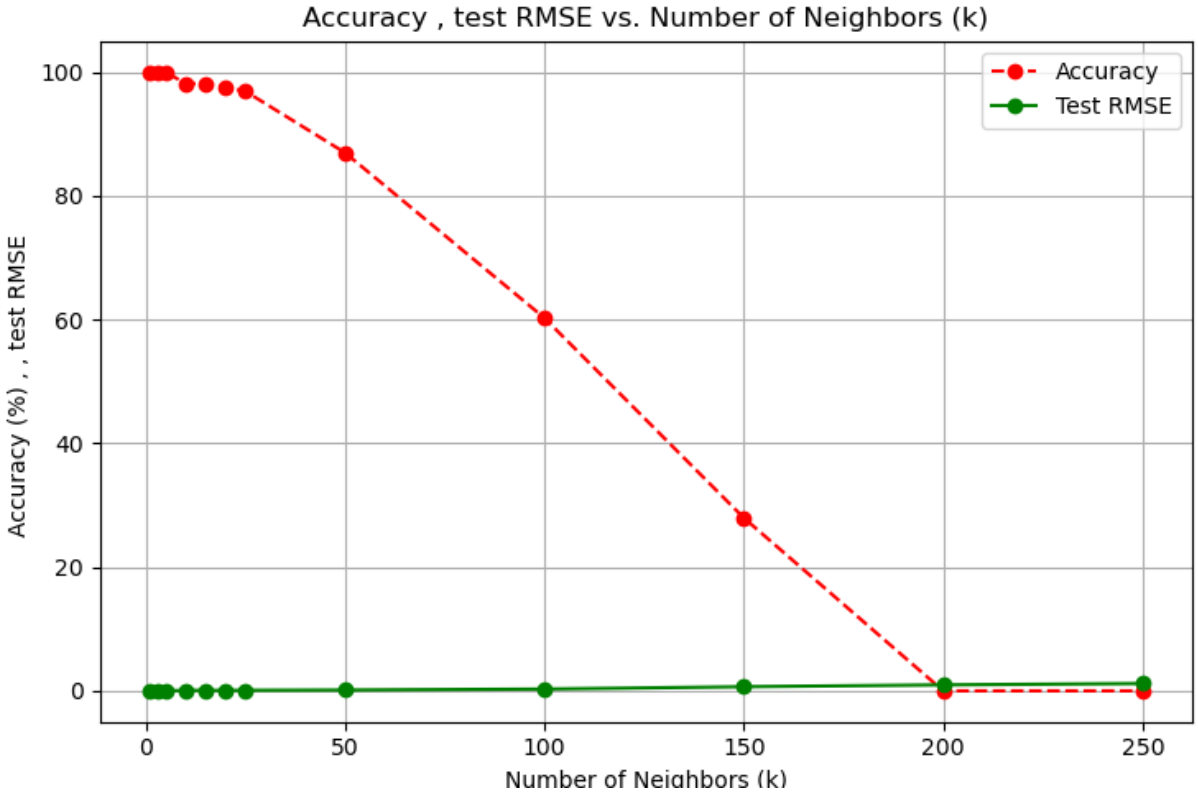
```
   Sex  Length  Diameter  Height  Whole_weight  Shucked_weight  Viscera_weight  Shell_weight
0   M   0.455     0.365   0.095        0.5140          0.2245          0.1010         0.150
1   M   0.350     0.265   0.090        0.2255          0.0995          0.0485         0.070
2   F   0.530     0.420   0.135        0.6770          0.2565          0.1415         0.210
3   M   0.440     0.365   0.125        0.5160          0.2155          0.1140         0.155
4   I   0.330     0.255   0.080        0.2050          0.0895          0.0395         0.055
5   I   0.425     0.300   0.095        0.3515          0.1410          0.0775         0.120

    Rings
0      15
1       7
2       9
3      10
4       7
5       8
Length            0.556720
Diameter          0.574660
Height            0.557467
Whole_weight      0.540390
Shucked_weight    0.420884
Viscera_weight    0.503819
Shell_weight      0.627574
Rings             1.000000
Name: Rings, dtype: float64
[1771 2060 2518]
[[8]
 [9]
 [8]]
predicted number of rings k=3, prediction=8.333333333333334

Predicted rings, using mode with k = 3, # rings=[8]

[1771 2060 2518  969 3726 1773 3728]
[[ 8]
 [ 9]
 [ 8]
 [ 8]
 [ 8]
 [10]
 [10]]
predicted number of rings k=7, prediction=8.714285714285714

Predicted rings, using mode with k = 7, # rings=[8]


number of neighbors used: 5
training error =  1.8494714777691794
Accuracy: 4.07%
testing error =  2.300551213332897
```

PROBLEM #02(CANCER TYPES) SOLUTION OUTPUT






Accuracy , train RMSE vs. Number of Neighbors (k)
Accuracy
Train RMSE
Accuracy (%) , , train RMSE
Number of Neighbors (k)
0
20
40
60
80
100
0
50
100
150
200
250


Accuracy , test RMSE vs. Number of Neighbors (k)
Accuracy
Test RMSE
Accuracy (%) , , test RMSE
Number of Neighbors (k)
0
20
40
60
80
100
0
50
100
150
200
250

```
shape:
(801, 20531)

data:
[[ 0.          2.01720929  3.26552691 ...  8.92132623  5.28675919
   0.        ]
 [ 0.          0.59273209  1.58842082 ...  9.39785429  2.09416849
   0.        ]
 [ 0.          3.51175898  4.32719872 ... 10.09046974  1.68302267
   0.        ]
 ...
 [ 0.          3.24958187  3.70749166 ...  9.46671072  4.6774575
   0.5866927 ]
 [ 0.          2.59033853  2.78797567 ... 10.40058062  5.71875068
   0.        ]
 [ 0.          2.32524248  3.80593214 ...  9.84479363  4.55071601
   0.        ]]

true labels:
['PRAD' 'LUAD' 'PRAD' 'PRAD' 'BRCA']

number of neighbors used: 5
training error =  0.10606601717798213
Accuracy: 100.00%
testing error =  0.0

number of neighbors used: 1
training error =  0.0
Accuracy: 100.00%
testing error =  0.0

number of neighbors used: 3
training error =  0.08838834764831845
Accuracy: 100.00%
testing error =  0.0

number of neighbors used: 5
training error =  0.10606601717798213
Accuracy: 100.00%
testing error =  0.0

number of neighbors used: 10
training error =  0.11812334654927 45
Accuracy: 98.14%
testing error =  0.08377728534855179

number of neighbors used: 15
training error =  0.12377174331988884
Accuracy: 98.14%
testing error =  0.07810735653085057
```

CONCLUSION

The implementation of the KNN algorithm on both datasets demonstrates promising predictions and accuracy, particularly in the Cancer dataset when compared to the Abalone dataset. Graphical representations of accuracy and errors versus the number of neighbors reveal notable differences between the two datasets.

However, it is worth noting that, across multiple trials with varying numbers of neighbors, neither the accuracy nor the mean consistently improves. For example, in the Cancer dataset, the accuracy significantly declines, nearing zero, when utilizing 200 as the number of nearest neighbors. This observation suggests that the choice of neighbor number plays a crucial role in the KNN algorithm's performance.