Cameron Lee

3/5/2023

Professor Rohit Kate

COMPSCI 411 - Assignment 1

Dataset 1 – Internet-Advertisement (ID Number: 40978)

Overview: This dataset is a compilation of possible image advertisements on the internet webpages.

Task: The task of this dataset is to find out whether an image is representative of an ad or not. This can be useful in identifying potential scam accounts via the internet. Additionally, this can be useful in finding out which advertisements on specific web pages generate the most money.

Target: The target of this dataset is either "noad" meaning the image is not an advertisement or "ad" meaning that the image on the webpage is an advertisement.

Features: Most of the features in this class are binary values that check whether or not a specific phrase appears in a website's URL. Some examples are origurl.contents.html, url.sunstrip.alley, and ancurl.exe.cid.

Dataset 2 – Bioresponse (ID Number: 4134)

Overview: This dataset is a compilation of molecules with their general characteristics to predict whether a specific biological response is elicited.

Task: The task of this dataset is to figure out the biological responses of molecules from their chemical properties. This is crucial in understanding elemental science and a molecule's general composition. Additionally, it is important to note that biological responses must be researched to provide a safe environment for all beings to live in.

Target: The target of this dataset is either (1) meaning a biological response was elicited or (0) a biological response was not elicited.

Features: Most of the features are hidden under aliases such as "D1 – D257". Each of these features represent calculated properties that can show some characteristics of the molecule. Some examples of these are size, shape, or elemental constitution.

Subtask 1

Internet-Advertisement

Internet-Advertisement Test Scores:

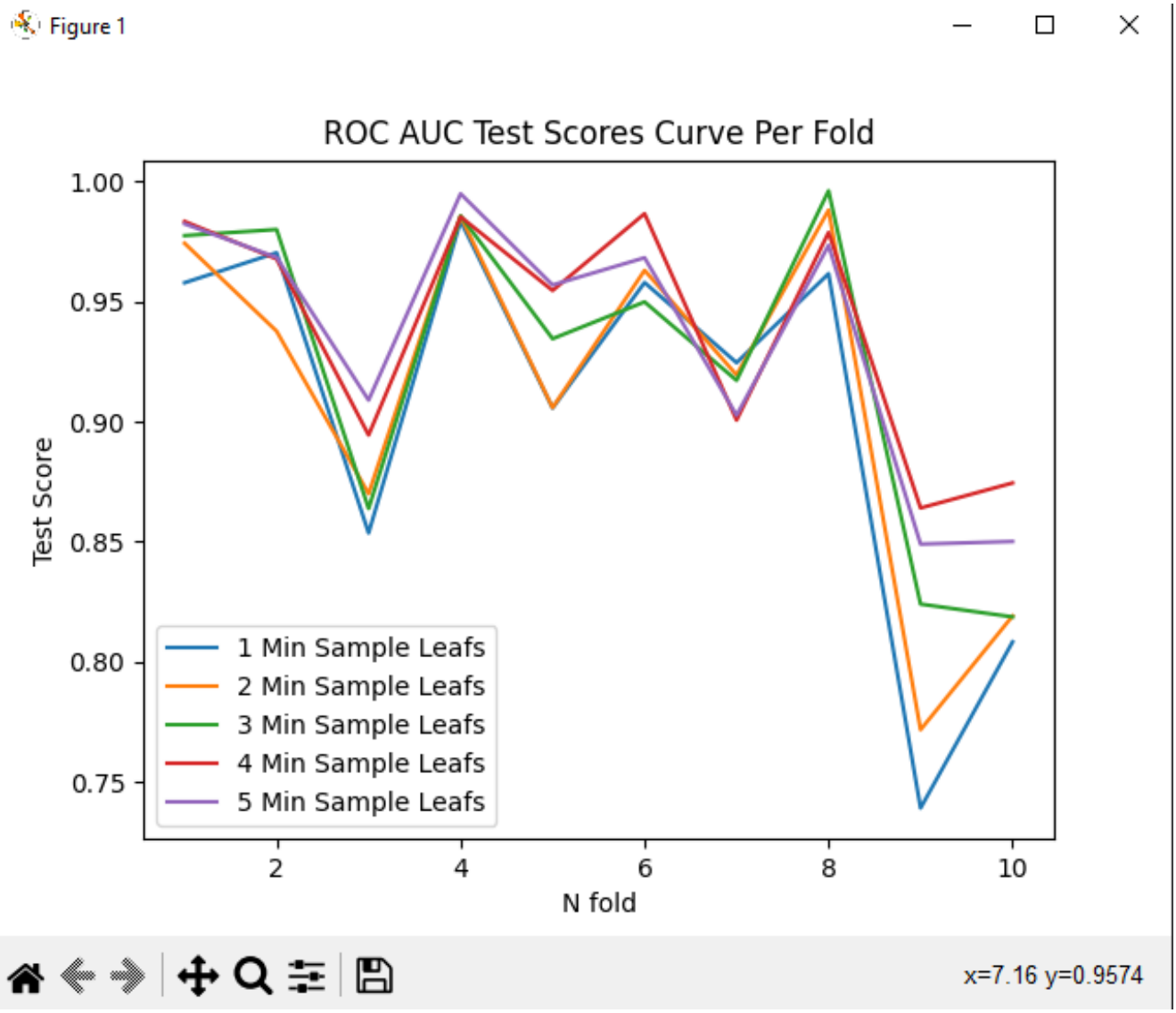

Figure: This figure above shows the varying ROC AUC test scores curve per fold per minimum number of sample leaves. This graph highlights where each decision tree had trouble or successfully predicted the training data in each individual fold. Additionally, the graph shows that a min_samples_leaf parameter of 1, 2, or 3 did not have a high scoring mean when combining all of the folds while 4 and 5 stayed consistently high throughout majority of the folds. According to the graph, each of decision trees struggled to make accurate predictions when reaching fold n= 9 or 10.

Internet-Advertisement Training Scores:

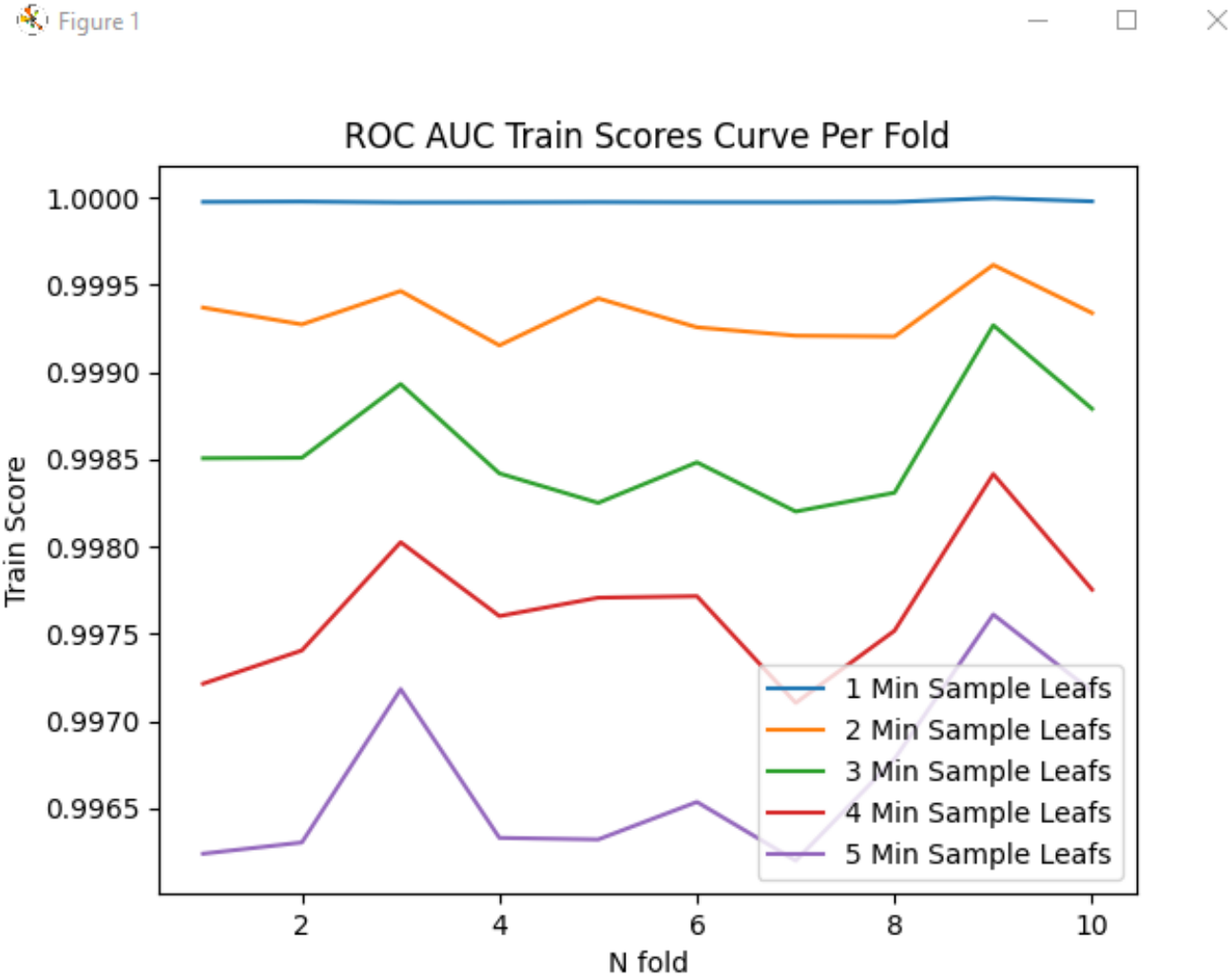

Figure: The figure above shows the ROC AUC training scores per fold per minimum sample leaves. This graph highlights the performance of training score per fold for each decision tree. It is important to note that a higher value of the min_samples_parameter is likely to achieve better performance on test data while worse on training data. So, when comparing this data to what is performing well on test data this is very accurate as the best parameters for the training data is 1, 2, and 3 while the worst are 4 and 5 minimum sample leaves. This clearly shows that there is an inverse relationship when doing cross-validation and calculating the test scores versus training scores.

Internet-Advertisement Mean ROC_AUC Test Scores Per Minimum Leaf Samples:

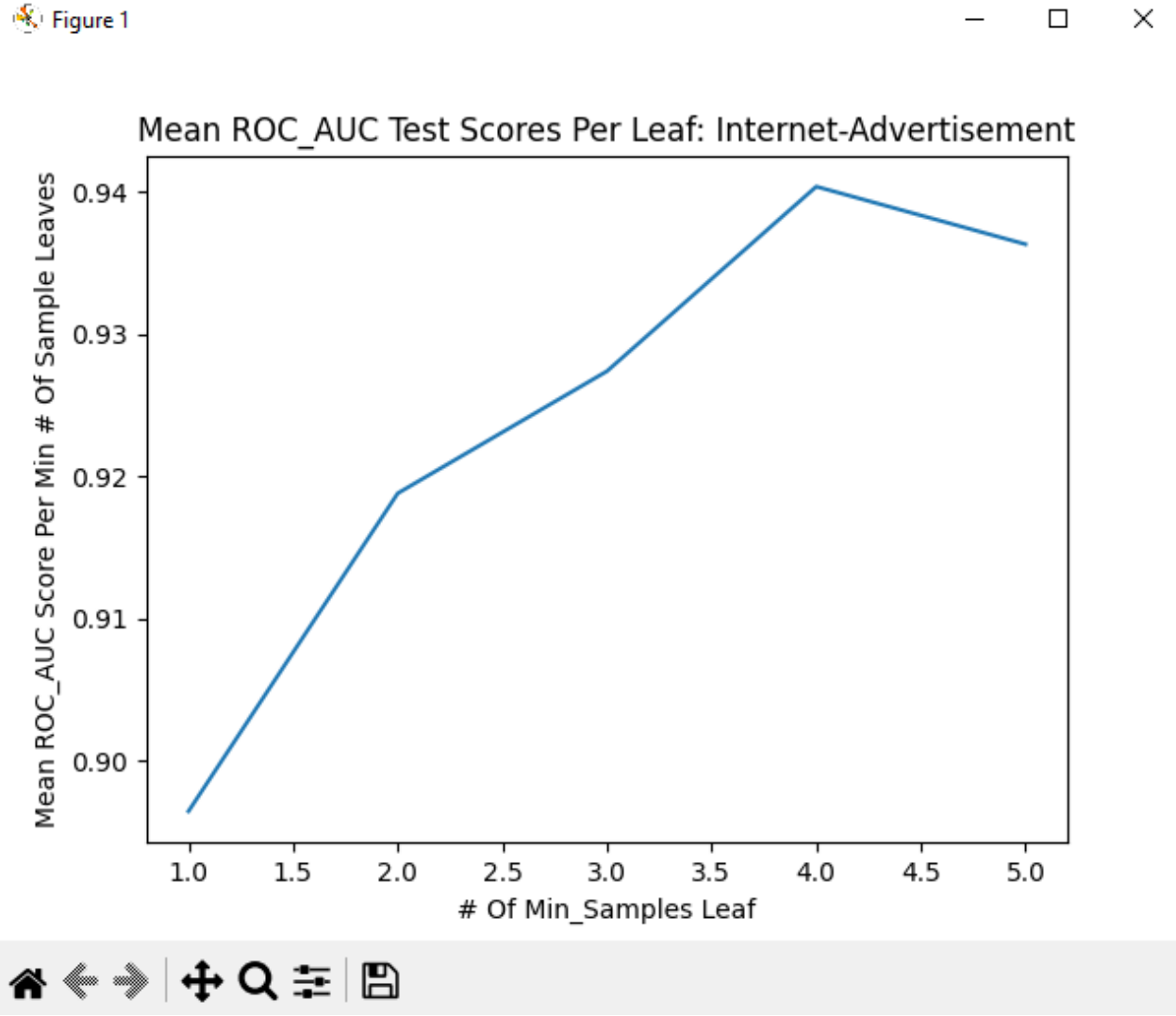

Figure: The figure above shows the Mean ROC_AUC test scores per leaf. This shows that there is **underfitting** when the number of minimum samples equals 5.0 since it crosses the imaginary true positive rate equal to the false positive rate bounded line. Additionally, the slight decrease in slope from min samples leaf [2.0-3.0] can be seen as a slight **overfitting** of the graph.

Mean ROC_AUC Test Scores Per Leaf (Bar Graph):

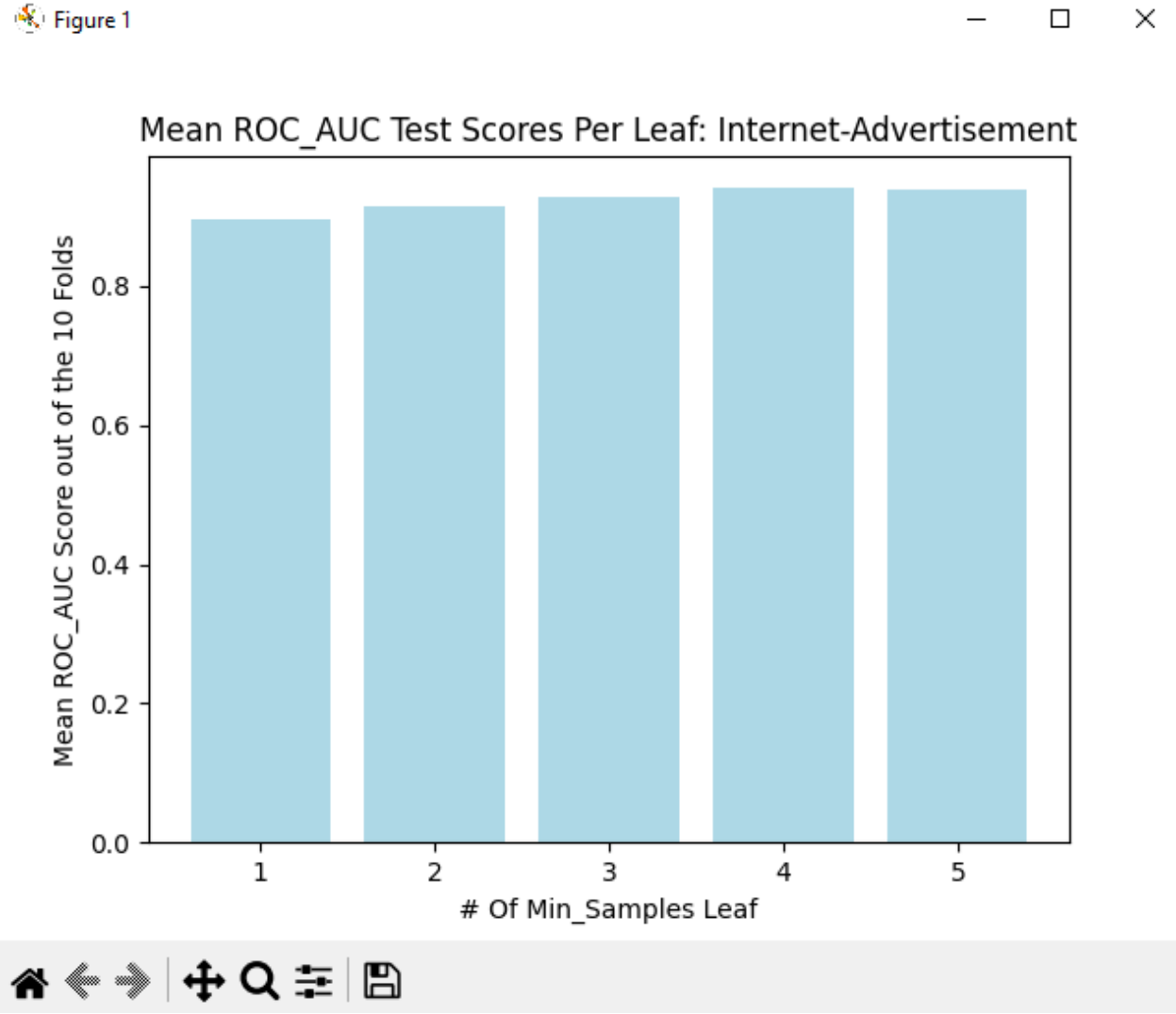

Figure: The graph above shows the Mean ROC_AUC test scores per number of minimum sample leaves. As this bar graph shows, the highest ROC_AUC score returned was from 4 minimum leaves closely followed by 5 minimum leaves. (To get exact numbers look at numbers in table below). As for the worst, parameters 1, 2, and 3 for the minimum number of sample leaves and 1 being the absolute worst parameter.

Mean ROC_AUC Test Scores Per Leaf (Table):

Minimum # Of Leaves	ROC_AUC Mean
1	0.8966773734881969
2	0.9141478226607738
3	0.9267564326583754
4	0.9406757289204097
5	0.9388553122965705

Bioresponse

Bioresponse Test Scores:

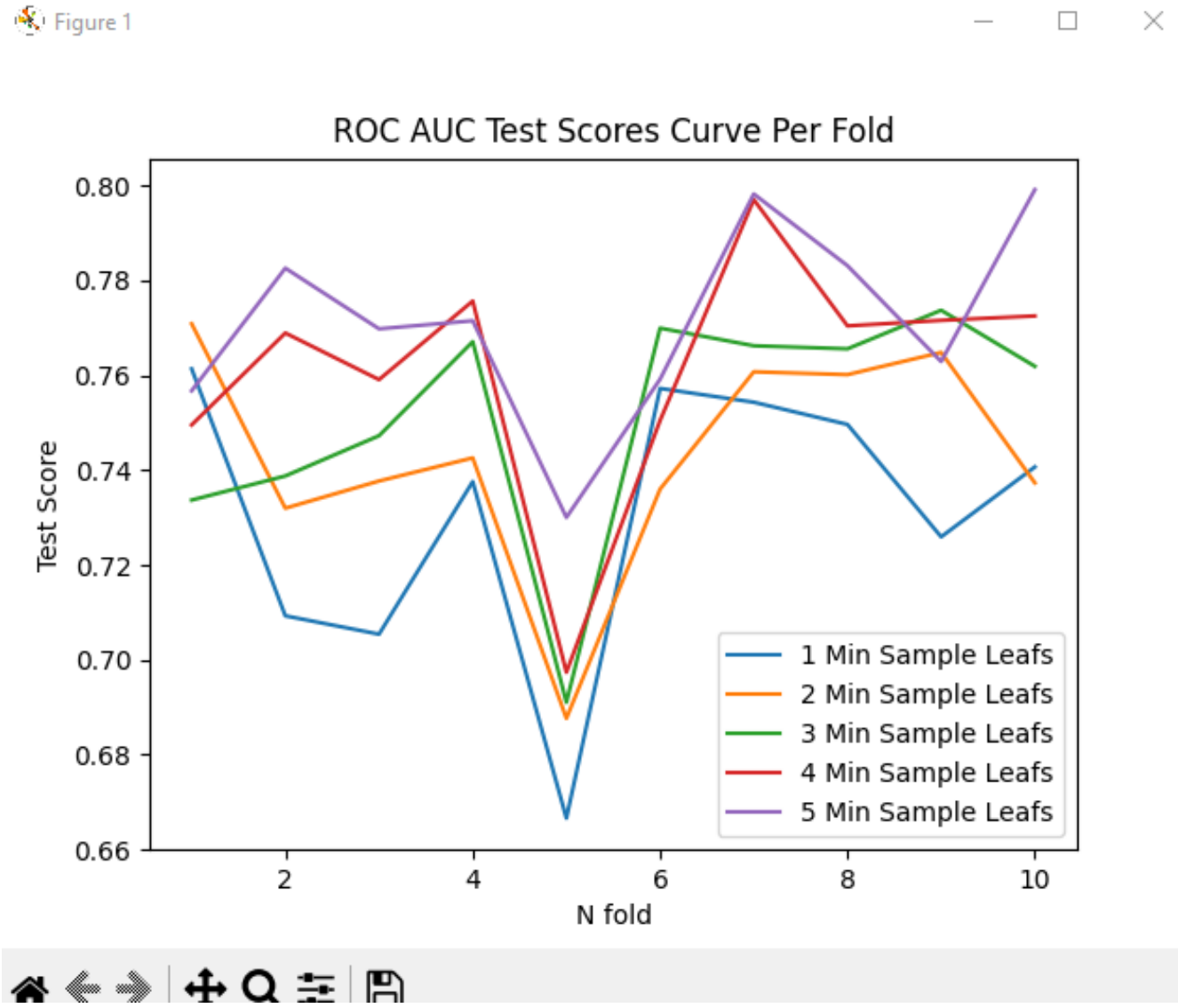

Figure: This figure above shows the varying ROC AUC test scores curve per fold per minimum number of sample leaves. This graph highlights where each decision tree had trouble or successfully predicted the training data in each individual fold. Additionally, the graph shows that a min_samples_leaf parameter of 1, 2, or 3 did not have a high scoring mean when combining all of the folds while 4 and 5 stayed consistently high throughout majority of the folds. According to the graph, each of decision trees struggled to make accurate predictions when reaching fold n = 5.

Bioresponse Training Scores:

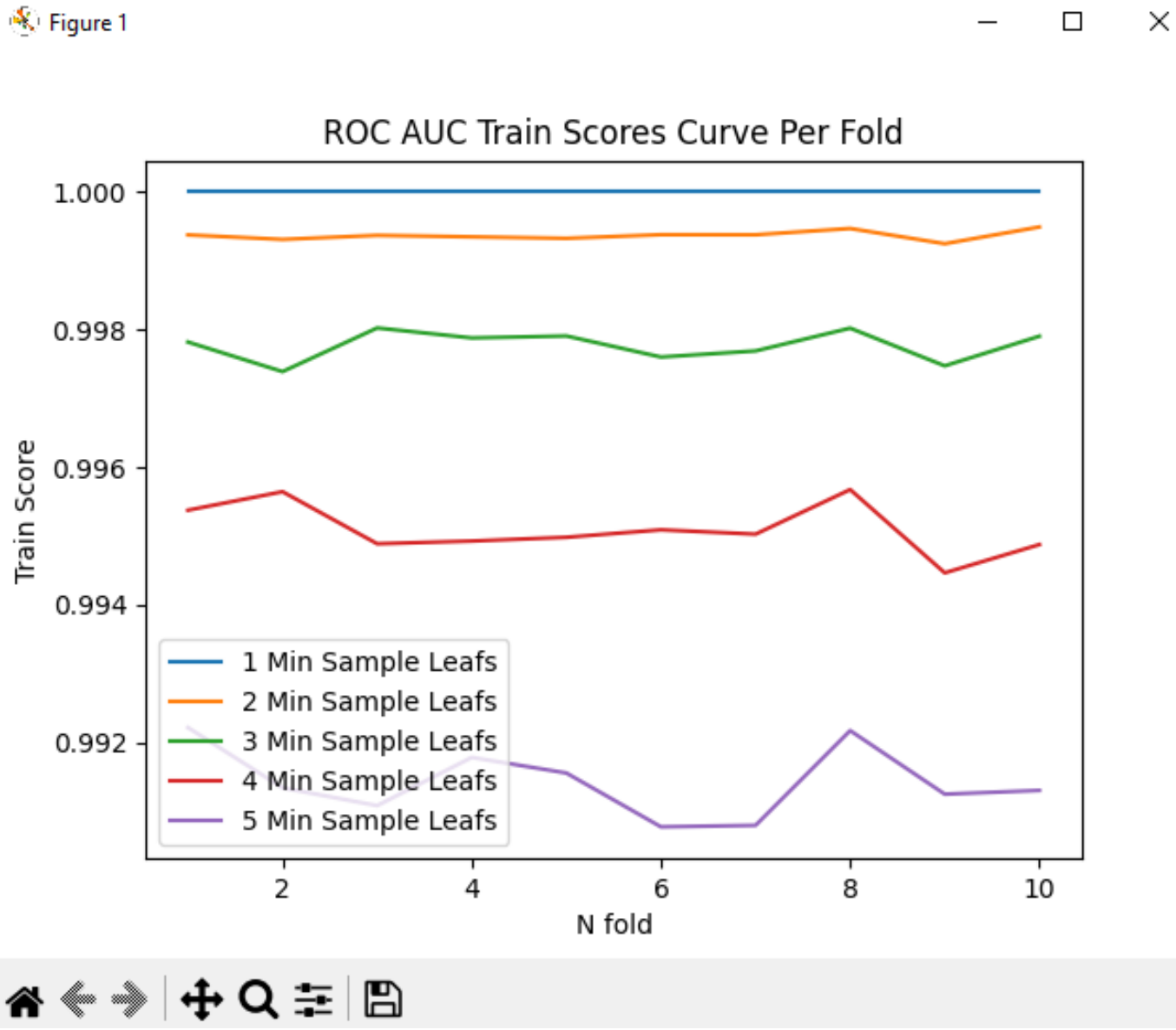

Figure: The figure above shows the ROC AUC training scores per fold per minimum sample leaves. This graph highlights the performance of training score per fold for each decision tree. It is important to note that a higher value of the min_samples_parameter is likely to achieve better performance on test data while worse on training data. So, when comparing this data to what is performing well on test data this is very accurate as the best parameters for the training data is 1, 2, and 3 while the worst are 4 and 5 minimum sample leaves. The absolute worst parameter being 5 minimum sample leaves. This clearly shows that there is an inverse relationship when doing cross-validation and calculating the test scores versus training scores.

Bioresponse Mean ROC_AUC Test Scores Per Leaf:

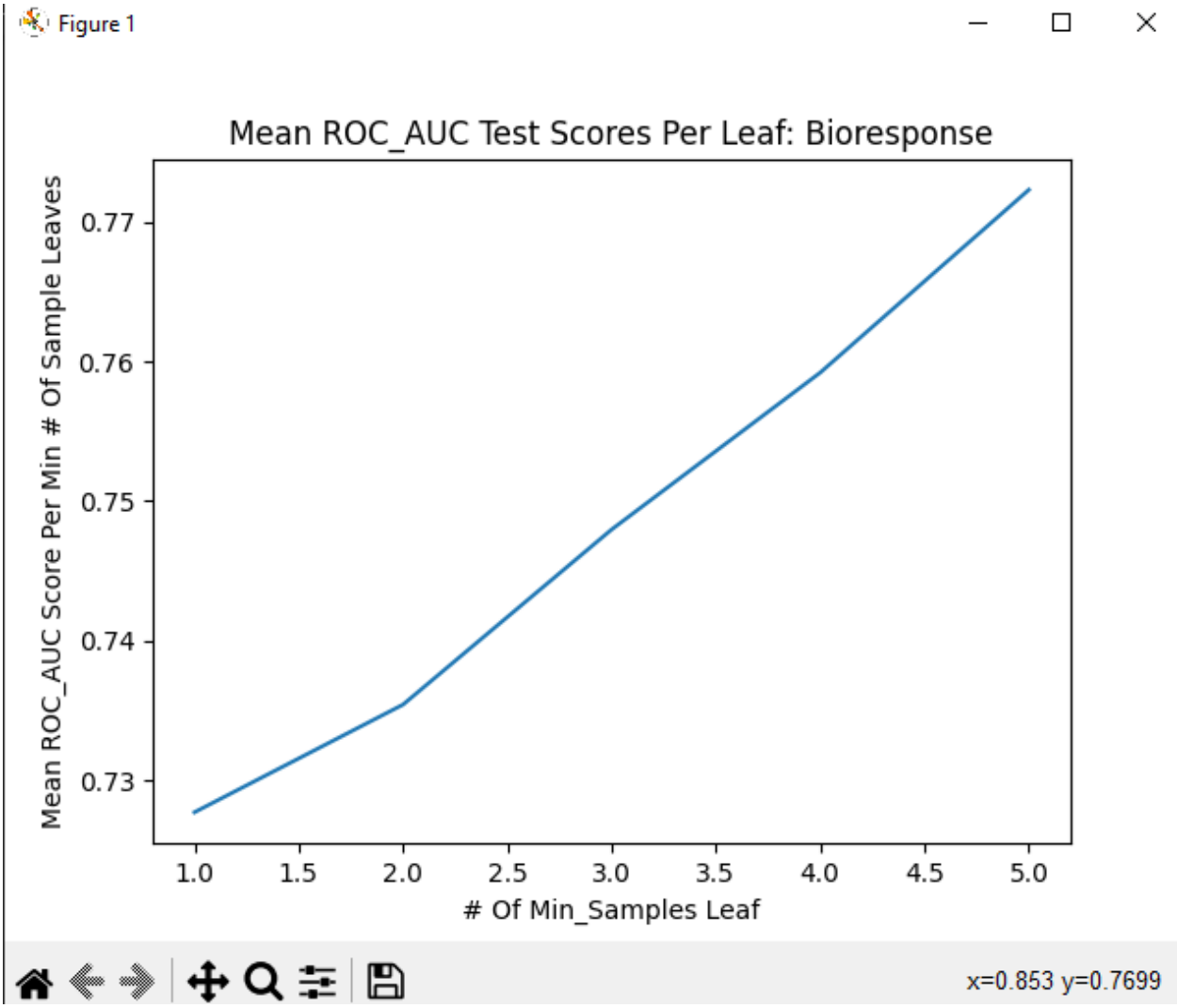

Figure: The figure above shows the Mean ROC_AUC test scores per leaf. This shows that there is **underfitting** throughout the entire predictions as the ROC_AUC score does not reach an average mean above 77%. This is not a good model overall for the dataset and requires a much larger parameter to allow for a better ROC_AUC score.

Mean ROC_AUC Test Scores Per Leaf (Bar Graph):

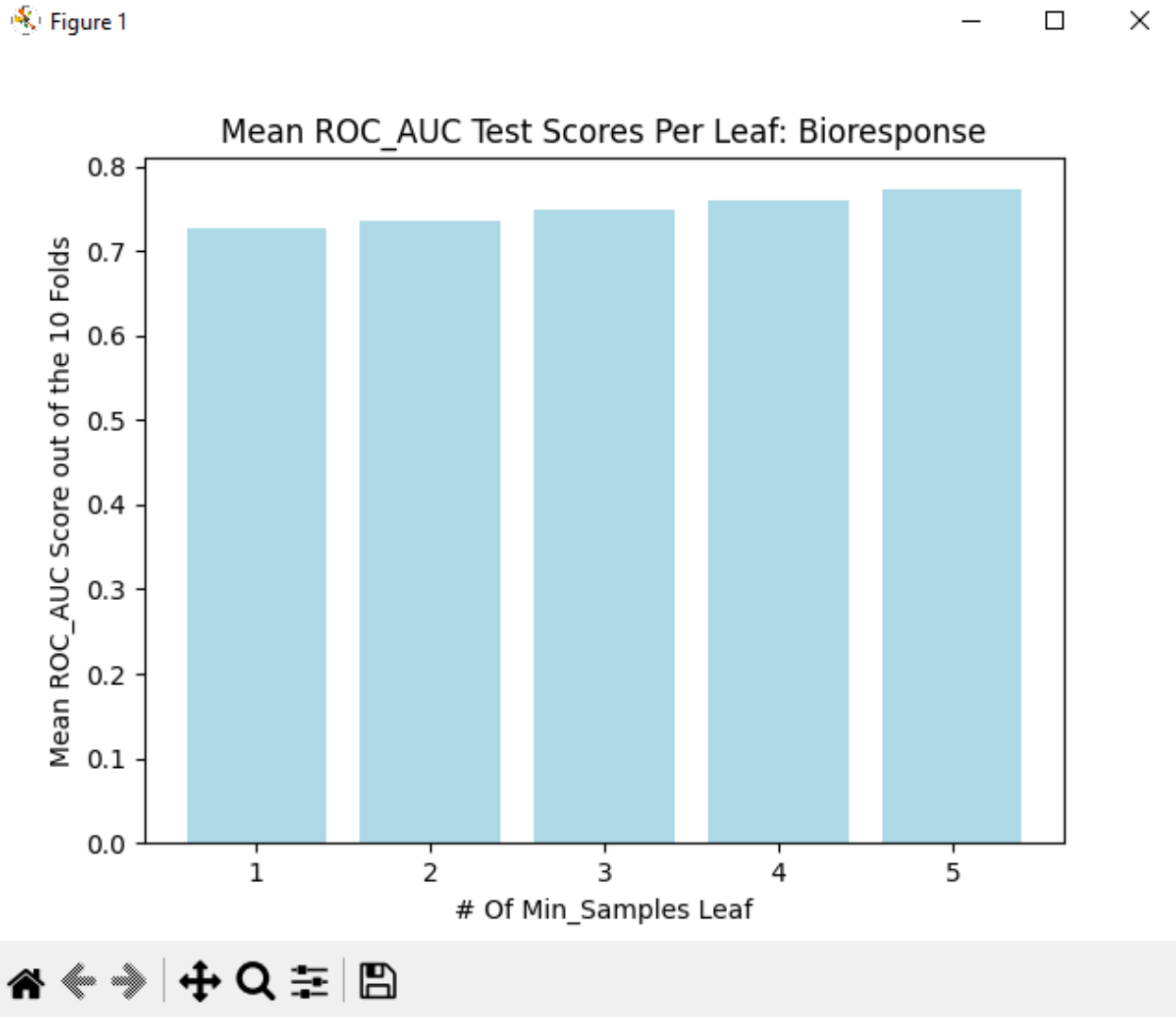

Figure: The graph above shows the Mean ROC_AUC test scores per number of minimum sample leaves. As this bar graph shows, the highest ROC_AUC score returned was from 5 minimum leaves closely followed by 4 minimum leaves. (To get exact numbers look at numbers in table below). As for the worst, parameters 1, 2, and 3 for the minimum number of sample leaves and 1 being the absolute worst parameter.

Mean ROC_AUC Test Scores Per Leaf (Table):

Minimum # Of Leaves	ROC_AUC Mean
1	0.7277166950152084
2	0.7354271306634146
3	0.7479623762763592
4	0.7591948887972114
5	0.7722821725636375

Subtask 2

Internet-Advertisement Prediction

Sample Parameters: [1-5]

```
========== RESTART: Shell ==========
import Assignment_1
Internet-Advertisement Best Parameter

Warning (from warnings module):
  File "C:\Users\Cameron Lee\AppData\Roaming\Python\Python311\site-packages\sklearn\datasets\_openml.p
    warn(
FutureWarning: The default value of `parser` will change from `'liac-arff'` to `'auto'` in 1.4. You ca
rtError` will be raised from 1.4 if the dataset is dense and pandas is not installed. Note that the pa
in fetch_openml's API doc for details.
After parameters
After dtc1
After grid search
The Test Score Mean of the Parameter Tuned Data is: 0.9348425668962209
The Training Score Mean of the Parameter Tuned Data is: 0.9972082136554379
The best parameter is: {'min_samples_leaf': 5}
```

Type Of Score Returned	Score Value
Test Score	0.9348425660962209
Training Score	0.9972082136554379

Using the GridSearchCV method to search for the best parameter gave a best parameter value of min_samples_leaf = 5. This means in this parameter list, using a parameter of 5 will yield the best results.

Since my results from earlier said that min_samples_leaf = 4 was the best value and was so close in roc_auc value to the min_samples_leaf = 5, I am confident that 5 is a very good prediction.

Additionally, since I tested only 1-5 sample leaves, I wanted to see what it would look like when comparing a larger range of parameters.

Sample Parameters: [2,4,6,8,10]

```
========== RESTART: Shell ==========
import Assignment_1
Internet-Advertisement Best Parameter

Warning (from warnings module):
  File "C:\Users\Cameron Lee\AppData\Roaming\Python\Python311\site-packages\sklearn\datasets\_openml.py", line 932
    warn(
FutureWarning: The default value of `parser` will change from `'liac-arff'` to `'auto'` in 1.4. You can set `parser='auto'` to silence this warning. Therefore, an `Impo
rtError` will be raised from 1.4 if the dataset is dense and pandas is not installed. Note that the pandas parser may return different data types. See the Notes Section
in fetch_openml's API doc for details.
After parameters
After dtc1
After grid search
The Test Score Mean of the Parameter Tuned Data is: 0.9333597149415835
The Training Score Mean of the Parameter Tuned Data is: 0.9935898556190675
The best parameter is: {'min_samples_leaf': 8}
```

Type Of Score Returned	Score Value
Test Score	0.9333597149515835
Training Score	0.9935898556190675

Using the GridSearchCV method to search for the best parameter gave a best parameter value of min_samples_leaf = 8. This means in this parameter list, using a parameter of 8 will yield the best results.

Bioresponse Prediction

Sample Parameters: [1-5]

```
import Assignment_1
Bioresponse Best Parameter

Warning (from warnings module):
  File "C:\Users\Cameron Lee\AppData\Roaming\Python\Python311\site-packages\sklearn\datasets\_openml.py", line 932
    warn(
FutureWarning: The default value of `parser` will change from `'liac-arff'` to `'auto'` in 1.4. You can set `parser='auto'` to silence this warning. Therefore, an `Impo
rtError` will be raised from 1.4 if the dataset is dense and pandas is not installed. Note that the pandas parser may return different data types. See the Notes Section
in fetch_openml's API doc for details.
After parameters
After dtc1
After grid search
The Test Score Mean of the Parameter Tuned Data is: 0.7834672327957994
The Training Score Mean of the Parameter Tuned Data is: 0.9839645575327591
The best parameter is: {'min_samples_leaf': 5}
```

Type Of Score Returned	Score Value
Test Score	0.7834672327957994
Training Score	0.9839645575327591

Using the GridSearchCV method to search for the best parameter gave a best parameter value of min_samples_leaf = 5. This means in this parameter list, using a parameter of 5 will yield the best results.

Since my results from earlier said that min_samples_leaf = 5, I am confident that 5 is a very good prediction.

Again, I decided to test a larger range of values.

Sample Parameters: [2,4,6,8,10]

```
========== RESTART: Shell ==========
import Assignment_1
Bioresponse Best Parameter

Warning (from warnings module):
  File "C:\Users\Cameron Lee\AppData\Roaming\Python\Python311\site-packages\sklearn\datasets\_openml.py", line 932
    warn(
FutureWarning: The default value of `parser` will change from `'liac-arff'` to `'auto'` in 1.4. You can set `parser='auto'` to silence this warning. Therefore, an `Imp
rtError` will be raised from 1.4 if the dataset is dense and pandas is not installed. Note that the pandas parser may return different data types. See the Notes Sectio
in fetch_openml's API doc for details.
After parameters
After dtc1
After grid search
The Test Score Mean of the Parameter Tuned Data is: 0.7918539661945431
The Training Score Mean of the Parameter Tuned Data is: 0.9567911934686808
The best parameter is: {'min_samples_leaf': 8}
```

Type Of Score Returned	Score Value
Test Score	0.7918539661945431
Training Score	0.9567911934686808

Using the GridSearchCV method to search for the best parameter gave a best parameter value of min_samples_leaf = 8. This means in this parameter list, using a parameter of 8 will yield the best results.