Question 1.

Evaluation metrics and evaluation metrics.

1. Implement ID3 Algorithm for Decision Tree discussed in class from scratch, the data can be downloaded from this link. You have to use entropy based Information Gain calculation method (as discussed in class) to evaluate different splits. Report Accuracy, Confusion Matrix and F1 Score in the report. What do you think is a good measure - Accuracy or F1 score, support your answer with proper claims.
2. Implement Decision Tree using sklearn library on the same data. (using gini index to calculate Information Gain).
3. Compare results of A and B, report the analysis.

Solution 1:

Disclaimer:

- If you want to run in CoLab you can click here:
- If you want to run the “MT19AIE321_Q1.py”, run like this:
    - `$python MT19AIE321_Q1.py <full_pathto_wifi_localization.txt>`
    - The code have the following dependencies(in terms of module)
        - pandas
        - numpy
        - matplotlib
        - sklearn
        - itertools
        - pprint
    - After the script is executed, it will produce the following two .png file in the present working directory
        - ROC_CURVE_SKLEARN_DT_Q2.png
        - ROC_CURVE_CUSTOM_DT_Q2.png

Report:

- This is in regard to the last run I had in my laptop
- Here is the Accuracy, Confusion Matrix and F1 for **Custom DT**

```
***************************************************************************************
1. Model Evaluation for 'CUSTOM DECISION TREE CLASSIFIER'
***************************************************************************************
Classification Report
              precision    recall  f1-score   support

           1       0.74      1.00      0.85        99
           2       1.00      0.86      0.93       101
           3       0.96      0.86      0.91       107
           4       0.99      0.88      0.93        93

    accuracy                           0.90       400
   macro avg       0.92      0.90      0.90       400
weighted avg       0.92      0.90      0.90       400
***************************************************************************************
```

- Accuracy or F1 score- what is a good measure ?
    - As we know, this depends lot of the problem statement, like Accuracy it is the measure of all the correctly (i.e. TPs and TNs) and F1 Score is the harmonic mean of Precision and Recall, which gives a good measure of the incorrectly classified class.
    - Now, in this problem, we were given with only the "csv" file, without any context, we do not have any idea about what each feature denotes. So, its hard to say which one(F1 score or Accuracy) would be more meaningful.
    - If we want feel that incorrect classification of each class label is critical, we should consider F1 score for each class label and we should focus on that more, and if correct classification is more important, we can just go with the Accuracy of the overall model
- Comparison of results from Custom DT Model and SKLearn DT Model
    - I observed the accuracy of the SKLearn more was better, also I noticed my custom model was slow and one of the reason was I went to the very depth of the tree, which is not very optimal
    - Confusion Matrix of Custom DT Model

```
- Confusion Matrix :
-  [[99  0  0  0]
-  [10 87  4  0]
-  [14  0 92  1]
-  [11  0  0 82]]
```

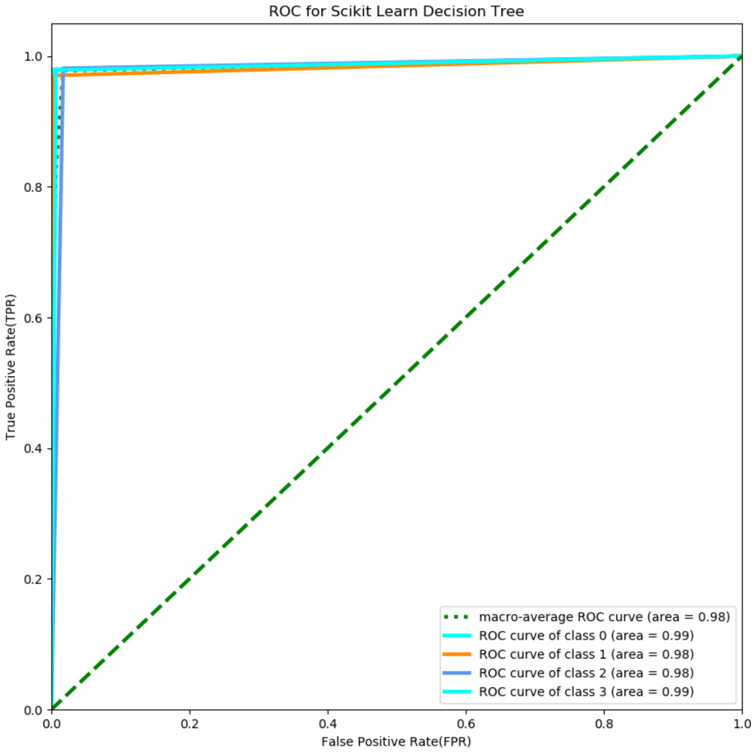
- Confusion Matrix for SKLear DT Model

```
- Confusion Matrix :
-  [[ 97   0   1   1]
-  [  0  98   3   0]
-  [  0   1 105   1]
-  [  1   0   1  91]]
```

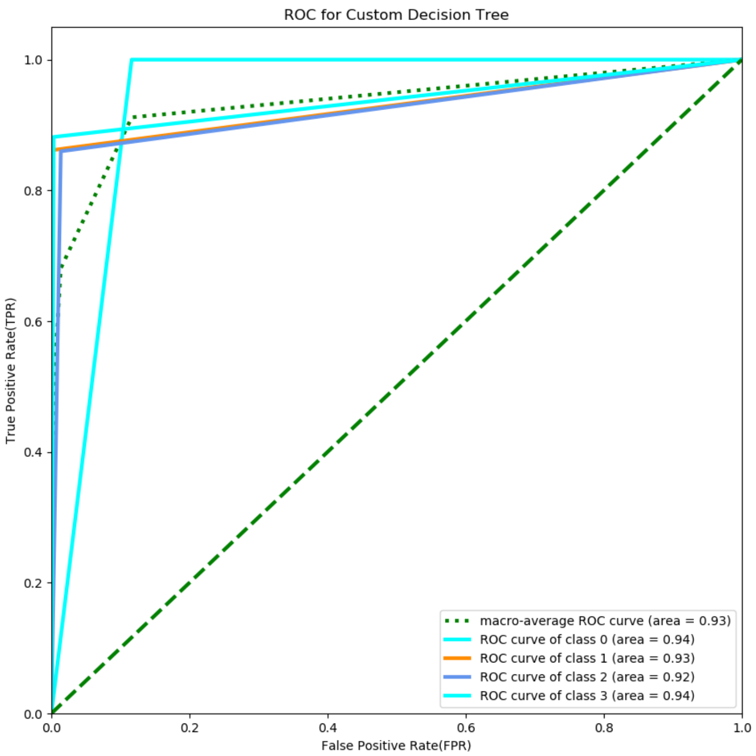
- Even the accuracy of the SKLearn DT Model was slightly better

```
2. Model Evaluation for 'SCI-KIT LEARN DECISION TREE CLASSIFIER'
*****************************************************************************************
Classification Report
              precision    recall  f1-score   support

           1       0.99      0.98      0.98        99
           2       0.99      0.97      0.98       101
           3       0.95      0.98      0.97       107
           4       0.98      0.98      0.98        93

    accuracy                           0.98       400
   macro avg       0.98      0.98      0.98       400
weighted avg       0.98      0.98      0.98       400


*****************************************************************************************
```

- ROC AUC Curve for SKLearn DT Model(this is better than Custom Decision Tree)



- ROC AUC Curve for **Custom Decision Tree**



Question 2.

Download the dataset from this link. Perform multi-class classification using Decision Trees and Random Decision Forest (RDF). You can use Sklearn library. You can perform hyperparameter tuning to improve your results and mention them in the report.

1. Report the following results on the training and test sets:
   (a) Accuracy
   (b) Confusion matrix
   (c) Precision and Recall
   (d) Sensitivity and Specificity
   (e) ROC curve (one for Decision tree and one for random forest)
   (f) Plot and visualize your decision tree (Only for Decision Tree, not Random Forest).

2. Compare and report accuracy achieved using Decision Trees and Random Forest on test set provided in question 2 above and suggest which classifier is better and why. Plot a single ROC curve corresponding to both the classifiers and compare their Area Under the Curve (AUC) scores and include them in your report.

Solution 2:

Disclaimer:

- If you want to run in CoLab you can [click here]:
- If you want to run the "MT19AIE321_Q2.py", run like this:
  - `$python MT19AIE321_Q2.py <full_path_to_iris_data_file>`
  - The code have the following dependencies(in terms of module)
    - pandas
    - numpy
    - matplotlib
    - sklearn
    - itertools
    - graphviz
  - After the script is executed, it will produce the following two .png file in the present working directory
    - ROC_CURVE_DT.png
    - ROC_CURVE_RF.png
    - ROC_DT_vd_RF_class_label_0.png
    - ROC_DT_vd_RF_class_label_1.png
    - ROC_DT_vd_RF_class_label_2.png
    - DT_Graph.pdf (Visual Tree)

Report:

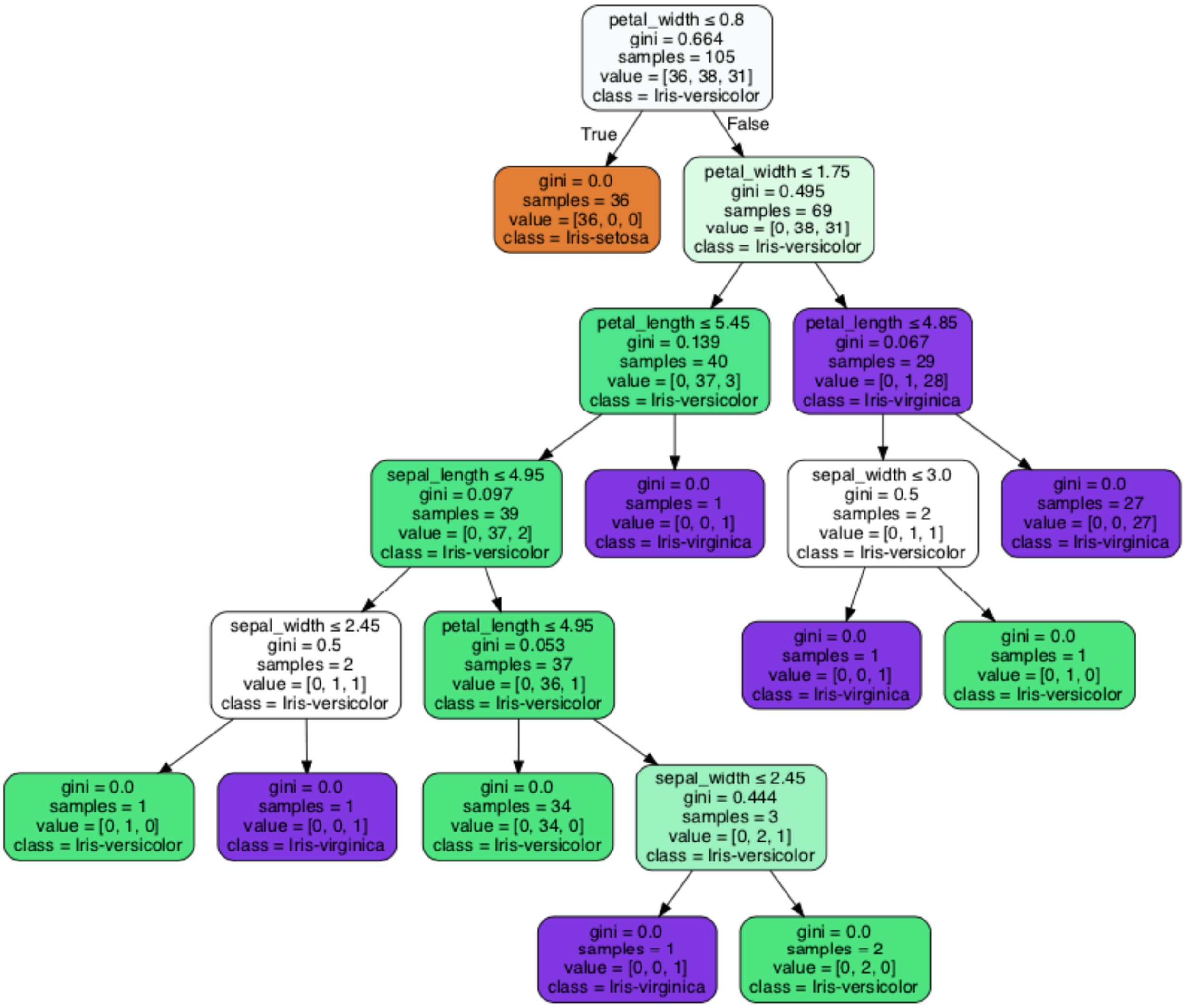
- This is in regard to the last run I had in my laptop
- Here are the performance matric with **Decision Tree Classifier**

```
***********************************************************************************************
Classification Report
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         9
           1       0.80      1.00      0.89         8
           2       1.00      0.85      0.92        13

    accuracy                           0.93        30
   macro avg       0.93      0.95      0.94        30
weighted avg       0.95      0.93      0.93        30

***********************************************************************************************
Accuracy : 0.9333333333333333
***********************************************************************************************
Confusion Matrix :
 [[ 9  0  0]
 [ 0  8  0]
 [ 0  2 11]]
***********************************************************************************************
Precision for class 0 : 1.0
Precision for class 1 : 0.8
Precision for class 2 : 1.0
***********************************************************************************************
Sensitivity for class 0 : 1.0
Sensitivity for class 1 : 1.0
Sensitivity for class 2 : 0.8461538461538461
***********************************************************************************************
Specificity for class 0 : 1.0
Specificity for class 1 : 0.9090909090909091
Specificity for class 2 : 1.0
***********************************************************************************************
```

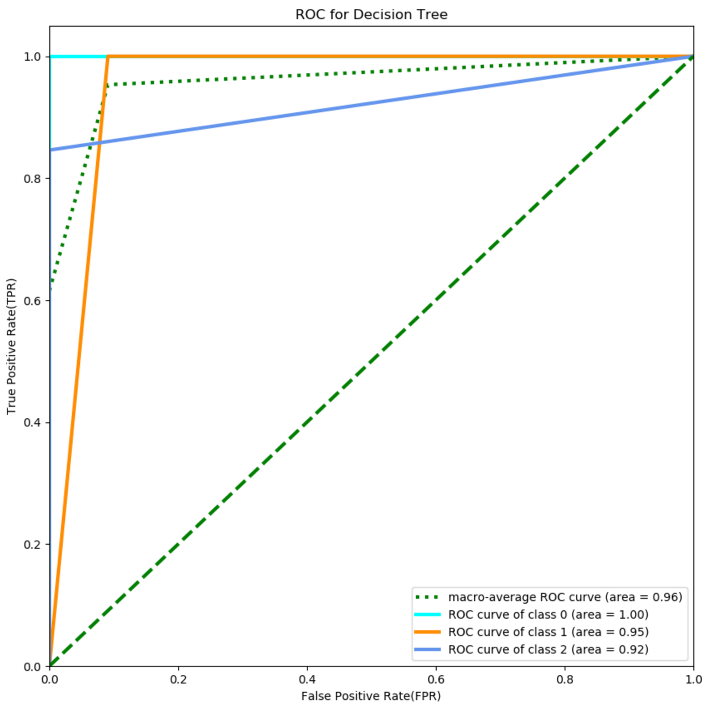
- Here are the performance matric with **Random Forest Classifier**

```
***********************************************************************************************
Classification Report
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         9
           1       0.89      1.00      0.94         8
           2       1.00      0.92      0.96        13

    accuracy                           0.97        30
   macro avg       0.96      0.97      0.97        30
weighted avg       0.97      0.97      0.97        30

***********************************************************************************************
Accuracy : 0.9666666666666667
***********************************************************************************************
Confusion Matrix :
 [[ 9  0  0]
 [ 0  8  0]
 [ 0  1 12]]
***********************************************************************************************
Precision for class 0 : 1.0
Precision for class 1 : 0.8888888888888888
Precision for class 2 : 1.0
***********************************************************************************************
Sensitivity for class 0 : 1.0
Sensitivity for class 1 : 1.0
Sensitivity for class 2 : 0.9230769230769231
***********************************************************************************************
Specificity for class 0 : 1.0
Specificity for class 1 : 0.9545454545454546
Specificity for class 2 : 1.0
***********************************************************************************************
```

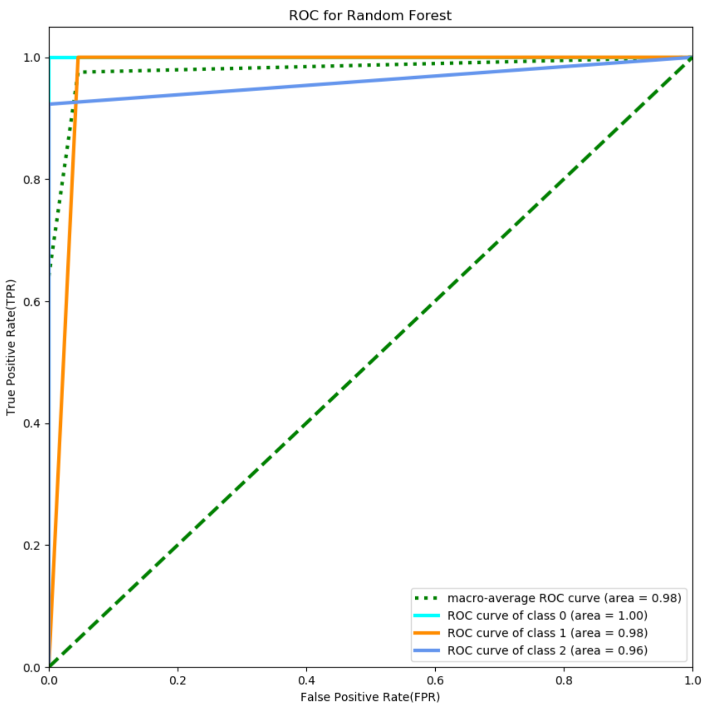
- As we can see the model performance for both the DT and RF is almost same, but RF performed little better
- I noticed that with different ration of training/test data split I got some different accuracy
- I also noticed if I change few of the hyper-parameters, like random_state, changing the criterion from gini to entropy, etc.
- For this problem I used all default.
- Plot and visualization of the **Decision Tree**



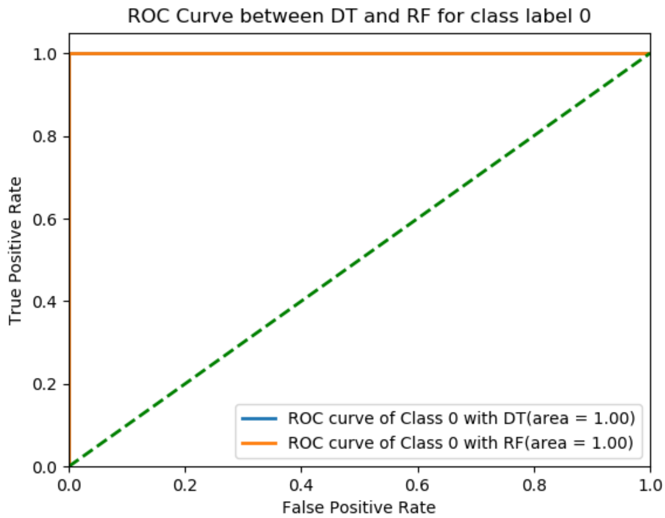
- ROC Curve for **Decision Tree**



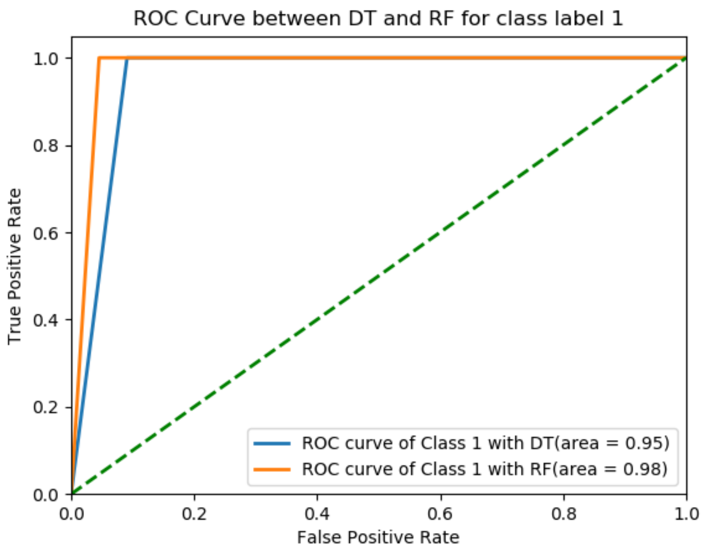
- ROC Curve for **Random Forest**



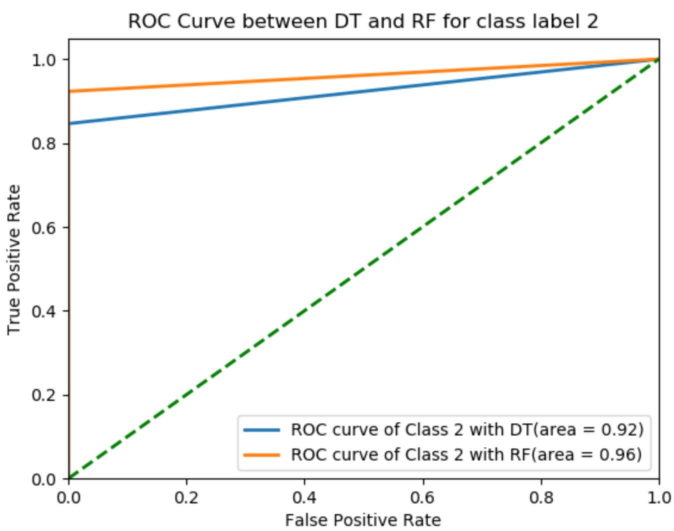
- ROC for Class 0 – DT vs RF



- ROC for Class 1 – DT vs RF



- ROC for Class 2 – DT vs RF



If we see this last plot for the class 2 label, we see that ROC AUC with Random Forest is more than Decision Tree. So, Random Forest performed better than Decision Tree