**Multiclass classifier decision tree using ID3 algorithm**

**Instruction on how to run my code:**

Please run this project on Google Collab and use the file named

“**decision\_tree\_ukm202.ipynb**” upload the files ‘iris.csv’ and ‘spambase.csv’ in the root folder. If uploaded in some other location, then please change the location yourself in the code by going in the last code block. After you open the file in google collab, please go to “Runtime->Run all”. By doing so, the program will ask you to [input.](../images/1.png) Please put **‘1’ for ‘iris.csv’** dataset or **‘2’ for ‘spambase.csv’** dataset. After giving the input you can see the result after the last code block.

**Result:**

Full result that is dataset before and after normalization, J, Leaf node for different set classes, Individual accuracy and cumulative accuracy across 10-cross validation for different values of nmin and standard deviation across 10-cross validation can be accessed by either running the program named “**decision\_tree\_with\_full\_result.ipynb**” or can be accessed from [here.](Full%20Result.docx)

I have intentionally done this to make the result more readable. On normal run, this is the result we will get:

|  |
| --- |
| Accuracy is 0.9333333333333333  Accuracy is 1.0  Accuracy is 1.0  Accuracy is 0.8666666666666667  Accuracy is 0.9333333333333333  Accuracy is 1.0  Accuracy is 1.0  Accuracy is 0.9333333333333333  Accuracy is 1.0  Accuracy is 1.0  Accuracy across 10-cross validation for 0.05 is 0.9666666666666668  Standard deviation across 10-cross validation for 0.05 is 0.04714045207910316  Accuracy is 1.0  Accuracy is 0.8666666666666667  Accuracy is 1.0  Accuracy is 1.0  Accuracy is 0.8666666666666667  Accuracy is 0.9333333333333333  Accuracy is 0.9333333333333333  Accuracy is 0.9333333333333333  Accuracy is 0.8666666666666667  Accuracy is 1.0  Accuracy across 10-cross validation for 0.1 is 0.9400000000000001  Standard deviation across 10-cross validation for 0.1 is 0.05837300238472753  Accuracy is 0.9333333333333333  Accuracy is 1.0  Accuracy is 0.9333333333333333  Accuracy is 1.0  Accuracy is 0.8666666666666667  Accuracy is 1.0  Accuracy is 1.0  Accuracy is 0.9333333333333333  Accuracy is 1.0  Accuracy is 0.9333333333333333  Accuracy across 10-cross validation for 0.15 is 0.9600000000000002  Standard deviation across 10-cross validation for 0.15 is 0.04661372658534006  Accuracy is 0.8666666666666667  Accuracy is 1.0  Accuracy is 0.8  Accuracy is 1.0  Accuracy is 0.8666666666666667  Accuracy is 1.0  Accuracy is 0.9333333333333333  Accuracy is 0.9333333333333333  Accuracy is 1.0  Accuracy is 0.9333333333333333  Accuracy across 10-cross validation for 0.2 is 0.9333333333333333  Standard deviation across 10-cross validation for 0.2 is 0.07027283689263064  Accuracy is 0.9333333333333333  Accuracy is 0.8666666666666667  Accuracy is 0.8666666666666667  Accuracy is 0.9333333333333333  Accuracy is 0.9333333333333333  Accuracy is 1.0  Accuracy is 1.0  Accuracy is 1.0  Accuracy is 1.0  Accuracy is 0.8666666666666667  Accuracy across 10-cross validation for 0.25 is 0.9400000000000002  Standard deviation across 10-cross validation for 0.25 is 0.05837300238472753 |

**Note1:** Please note that the result may vary on every run. This is because we are randomly shuffling the data and are using 10-cross validation.

**Note2 (About spambase dataset):** Please also note that the program will take too long to load the result for “**spambase.csv**” dataset. I run the program on GPU machine, and until 1800th row it gave me the result, however it gave accuracy 1.0 and here also it took too long beyond that dataset. I have provided all the results obtained from GPU machine in folder GPU output with trimmed spambase dataset if you may want to look and run the code. You can access them through these links:

* [Screenshot](../GPU_output/1.png) [1](../GPU_output/1.png) – taking too long for whole dataset
* [Screenshot 2](../GPU_output/2.png) – taking too long for 2000 rows
* [Screenshot 3](../GPU_output/3.png) – working for 1800, taking too long for 1950
* [Output 1](../GPU_output/output_1.pdf) – setup and result of iris dataset
* [Output 2](../GPU_output/output_2.pdf) – working for 1800 rows
* [Both output](../GPU_output/Both_output.pdf) – result of iris dataset and spambase dataset with 1800 rows

I have also provided the same code with **.py** extension named “**decision\_tree\_with\_py\_extn.py**” in case you would like to run on GPU machine.