Alper Canberk Balcı - 2017400087
20/12/2021

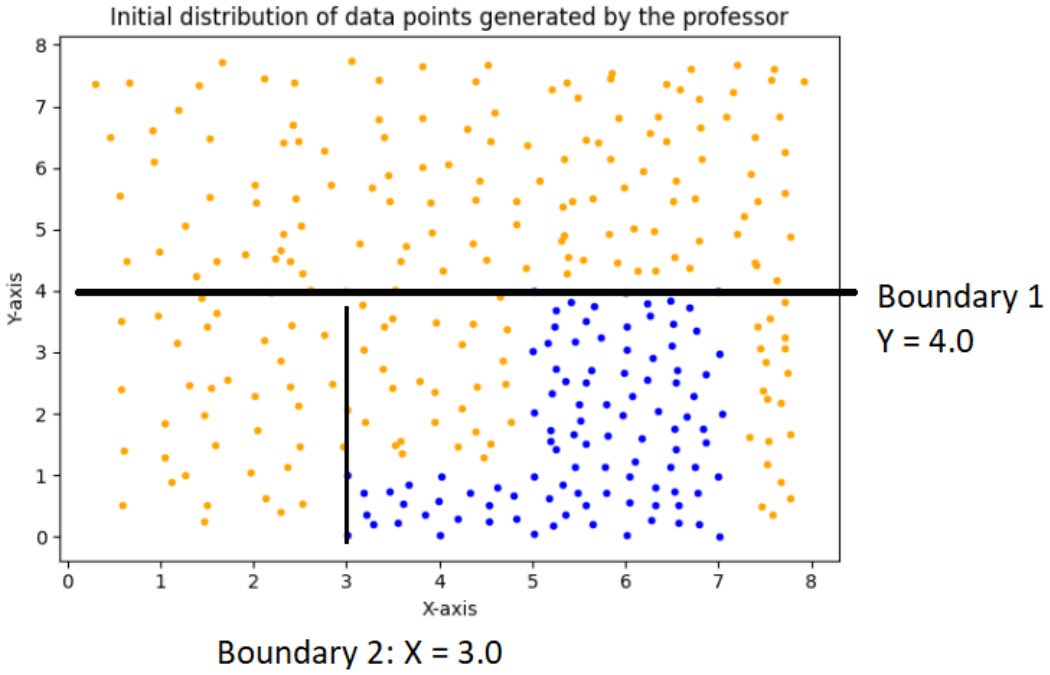
CMPE481 Data Analysis and Visualization Project-1

1) I have given a 2D dataset. I wrote the algorithm, and it works as if it is a built-in function, but I designed it for max_depth = 2 hyperparameter adjustment. There can be minimal modifications to change the algorithm to work with more max_depth. It gives results similar to the sklearn decision tree algorithm. My program prints the wanted values in the project description, as a report into the console. This is the output of my program when we run the Assignment2.py:

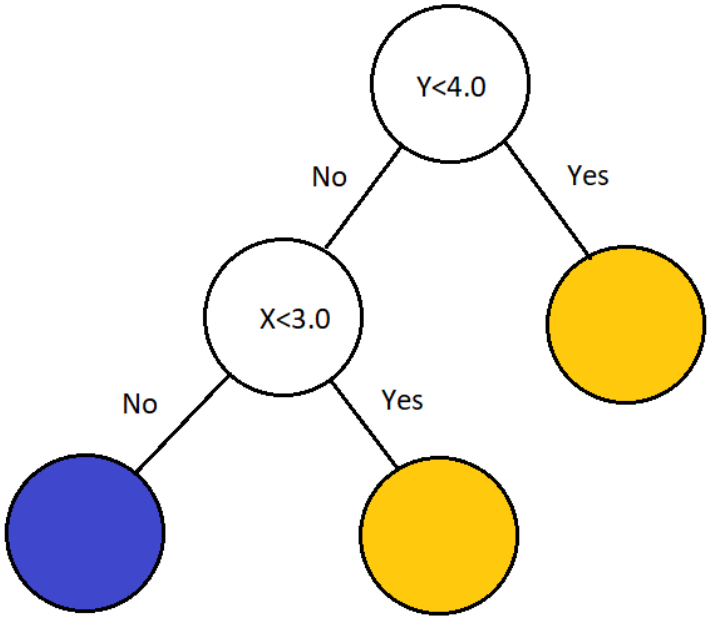
   level=1, axis=1 so, y = 4.00 is the root node
   level=2, axis=0 so, x = 3.00 is the level-2 node
   Printing the report:

   id: 0 level: 1 axis: Y boundary value 4.000883392226143 left, right, weighted entropys: [0.9940302114769565, 0.0, 0.595780928032663]
   id: 1 level: 2 axis: X boundary value 3.0012367491166048 left, right, weighted entropys: [0.0, 0.899349319724299, 0.71659384298888]
   Printing the report:

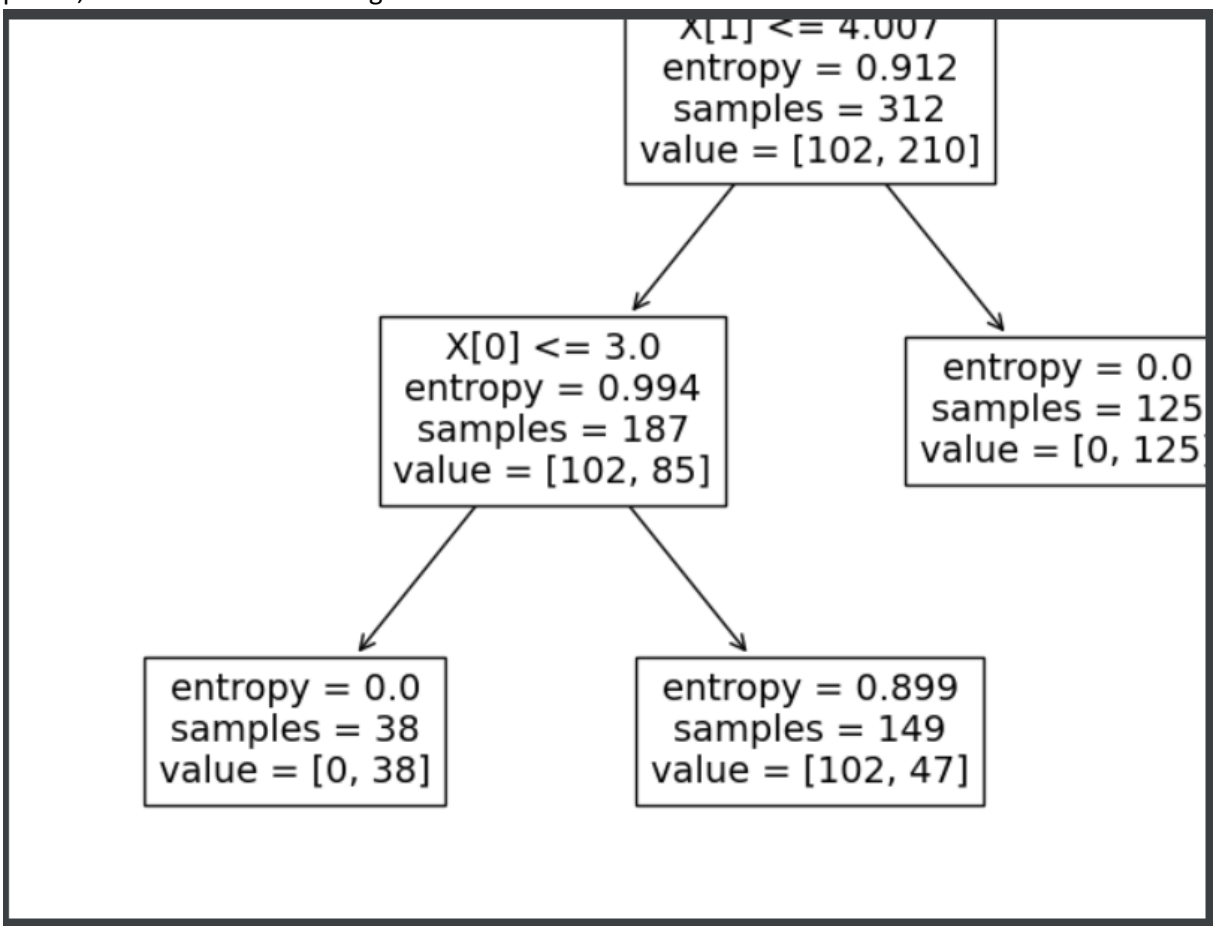
   id: 0 level: 1 axis: X boundary value 5.970302622559747 left, right, weighted entropys: [0.7588840483719566, 0.8879763195151351, 0.7898661934463195]
   id: 1 level: 2 axis: X boundary value 1.9203026225597608 left, right, weighted entropys: [0.9268190639645772, 0.0, 0.3089396879881924]
   id: 2 level: 2 axis: Y boundary value 4.016275710324431 left, right, weighted entropys: [0.0, 0.0, 0.0]
   Here we can see the first report is for the given dataset, and the second one is for the dataset I generated myself.

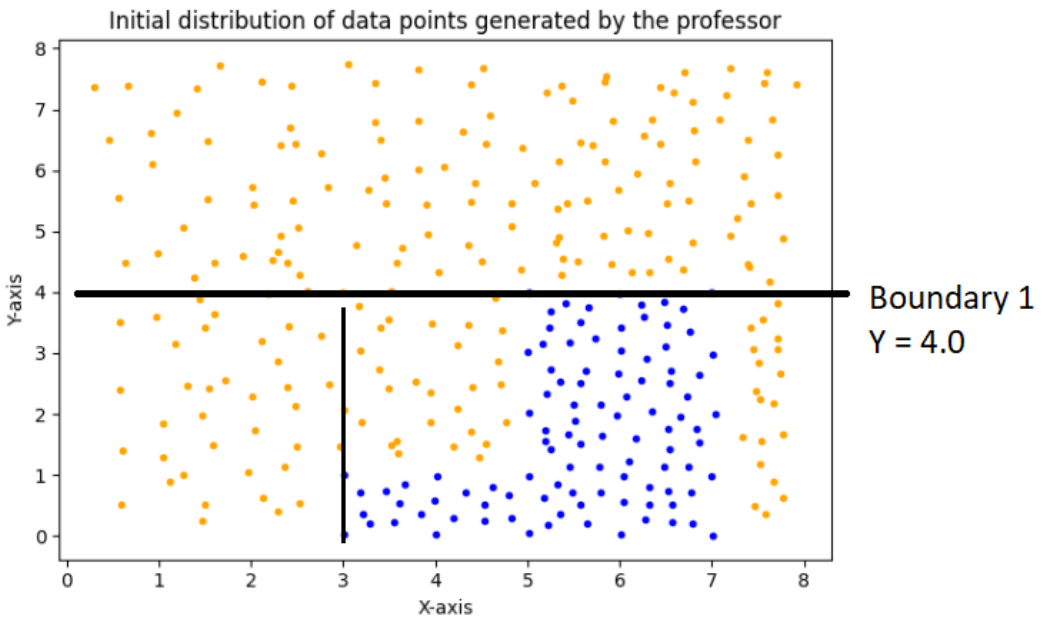


2) I drew the decision tree via Microsoft Paint.



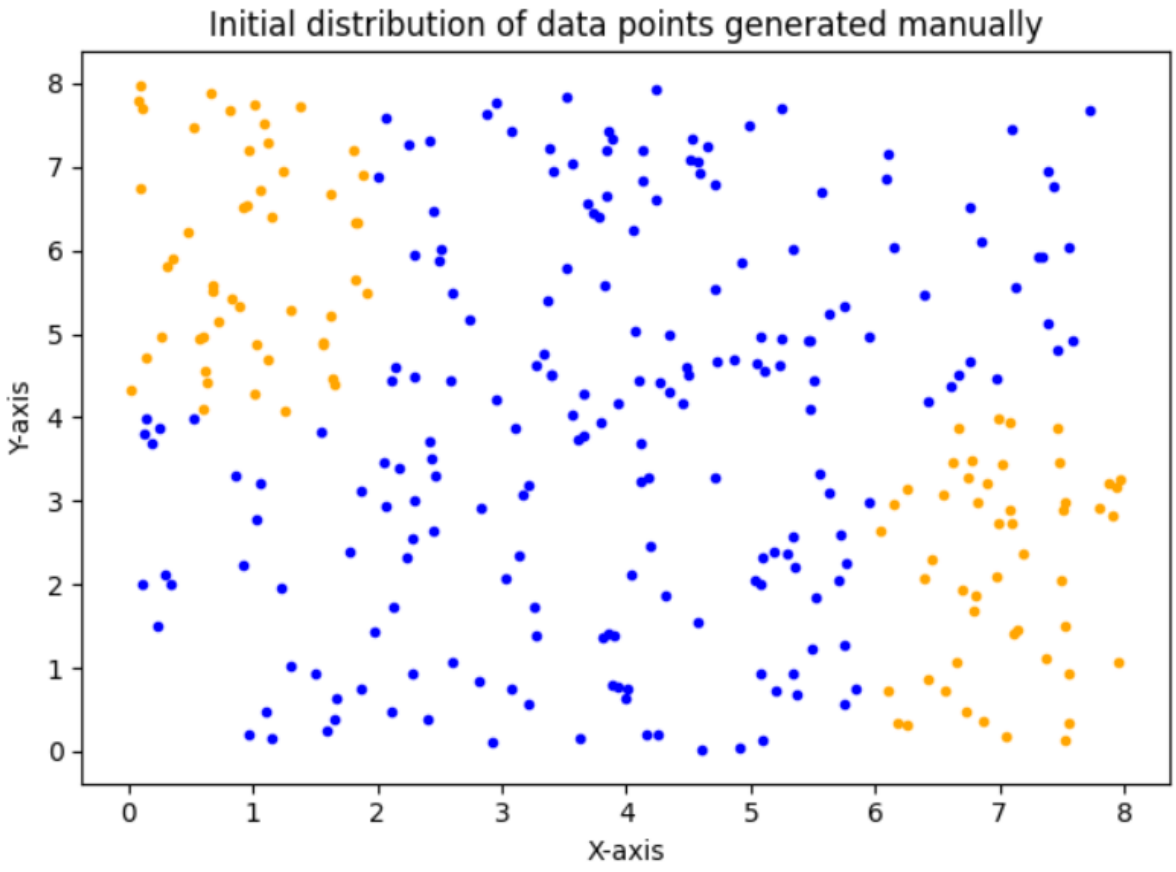
3) I I plotted the decision tree using tree.plot_tree(clf) function but the image's top part is corrupted, I don't know why. Here X[1] means y axis, and X[0] means x axis. When we compare the values, they are very close, I used 0.05 distance for iterating through the split points, that is where I was not generous. 0.007 difference in Y value comes from that.





Same figure is put here as the boundaries are found the same.

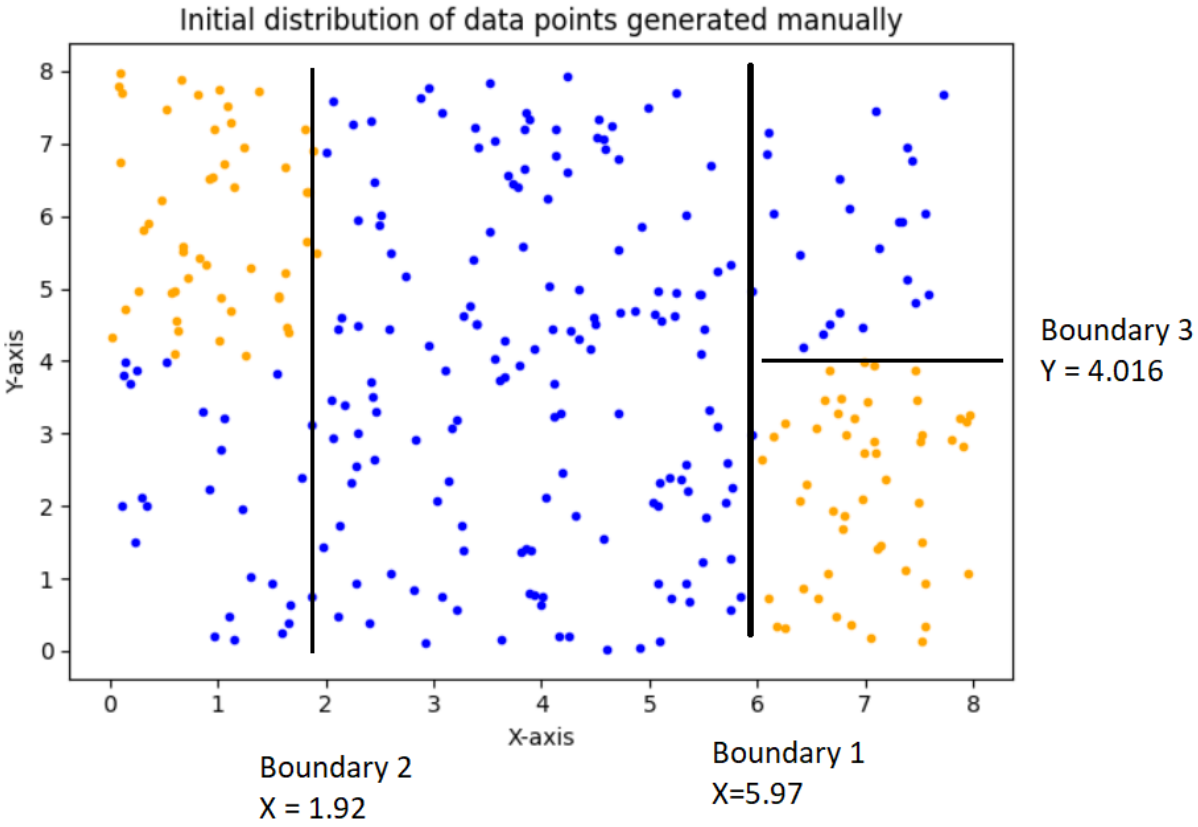
4) Did these same things for the 2D dataset that I created. I managed to generate a dataset just to have 3 decision boundaries, which is maximum in max_depth=2. The data set I created:

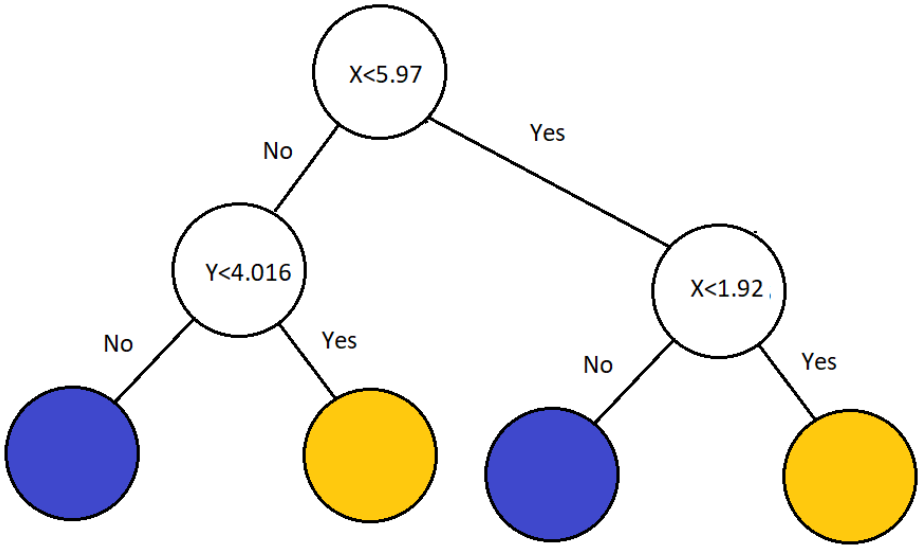


I applied firstly my algorithm. The values are in the part 1, actually:
Printing the report:

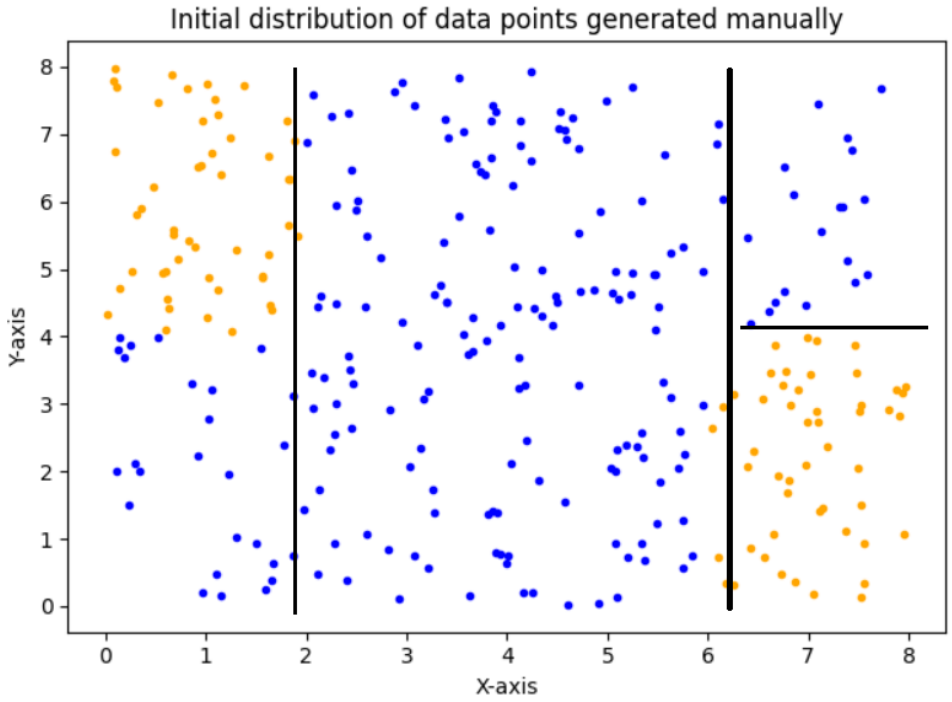
id: 0 level: 1 axis: X boundary value 5.970302622559747 left, right, weighted entropys: [0.7588840483719566, 0.8879763195151351, 0.7898661934463195]
id: 1 level: 2 axis: X boundary value 1.9203026225597608 left, right, weighted entropys: [0.9268190639645772, 0.0, 0.3089396879881924]
id: 2 level: 2 axis: Y boundary value 4.016275710324431 left, right, weighted entropys: [0.0, 0.0, 0.0]

here left and right is meant to be bottom and top, as they mean less and more.
The decision boundaries and decision tree I found are drawn here:





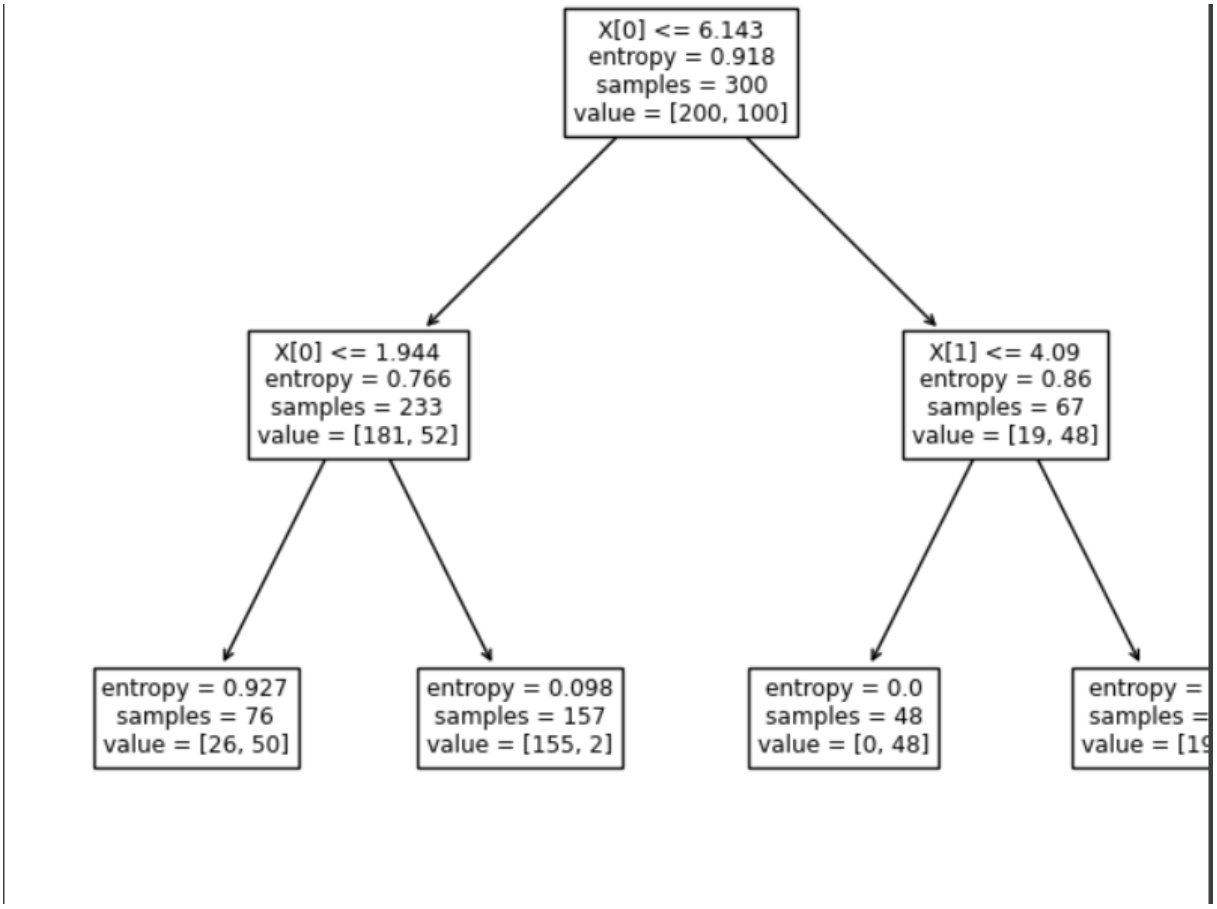
The decision boundaries and decision tree the sklearn algorithm found are here:



Boundary 3
Y = 4.09

Boundary 2
X = 1.944

Boundary 1
X = 6.143



The tree is again generated automatically. As we can see, they perform similarly. It is a complex dataset and hard to solve with max_depth=2, and they found similar results.