CS 6375

ASSIGNMENT 1

Names of students in your group:

Sai Vikas Meda: SXM190011

Sneha Elizabeth Sam: SES190004

Number of free late days used: 0

Note: You are allowed a **total** of 4 free late days for the **entire semester**. You can use at most 2 for each assignment. After that, there will be a penalty of 10% for each late day.

Please list clearly all the sources/references that you have used in this assignment.

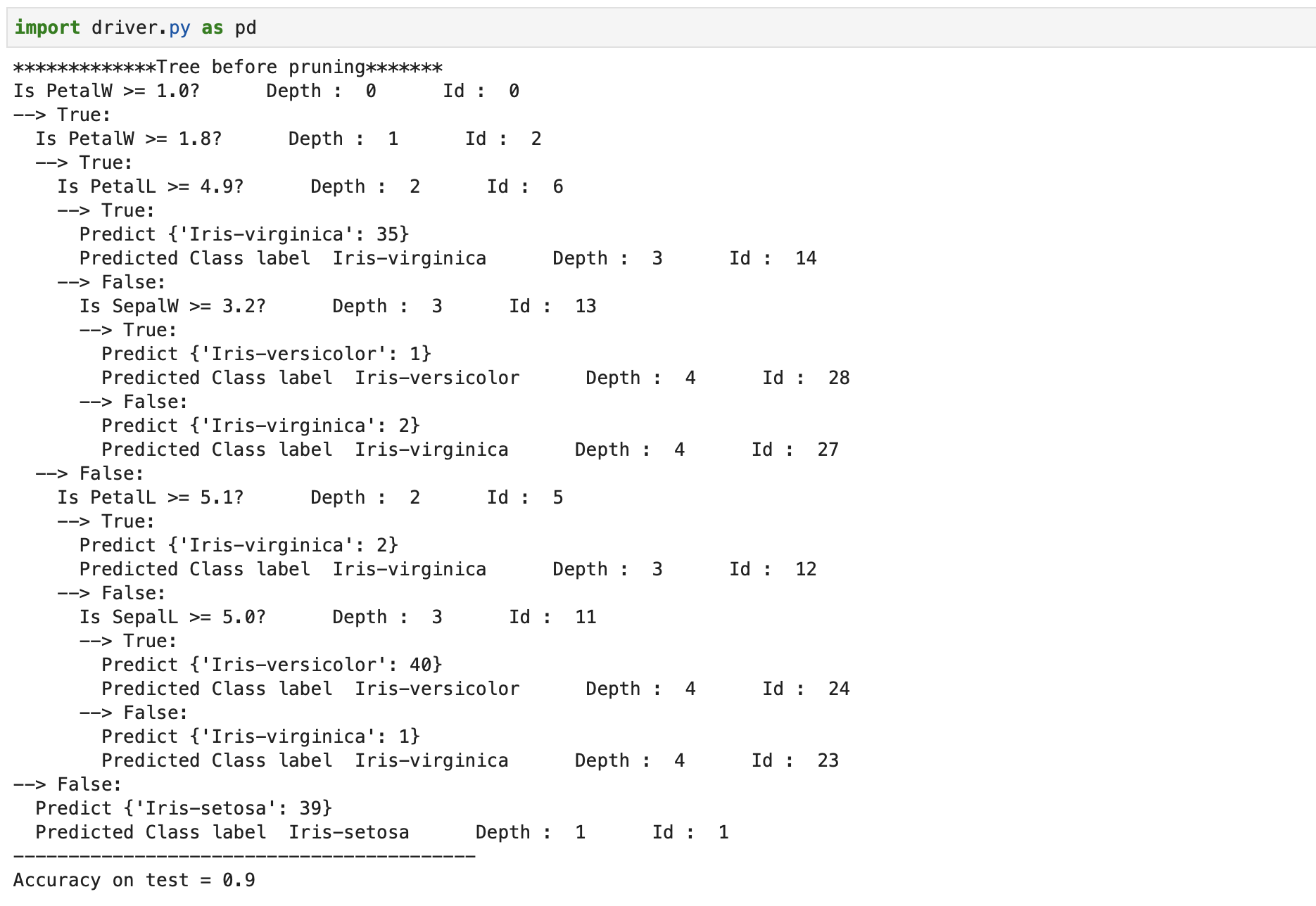
[www.google.com](http://www.google.com): For Python Syntax references

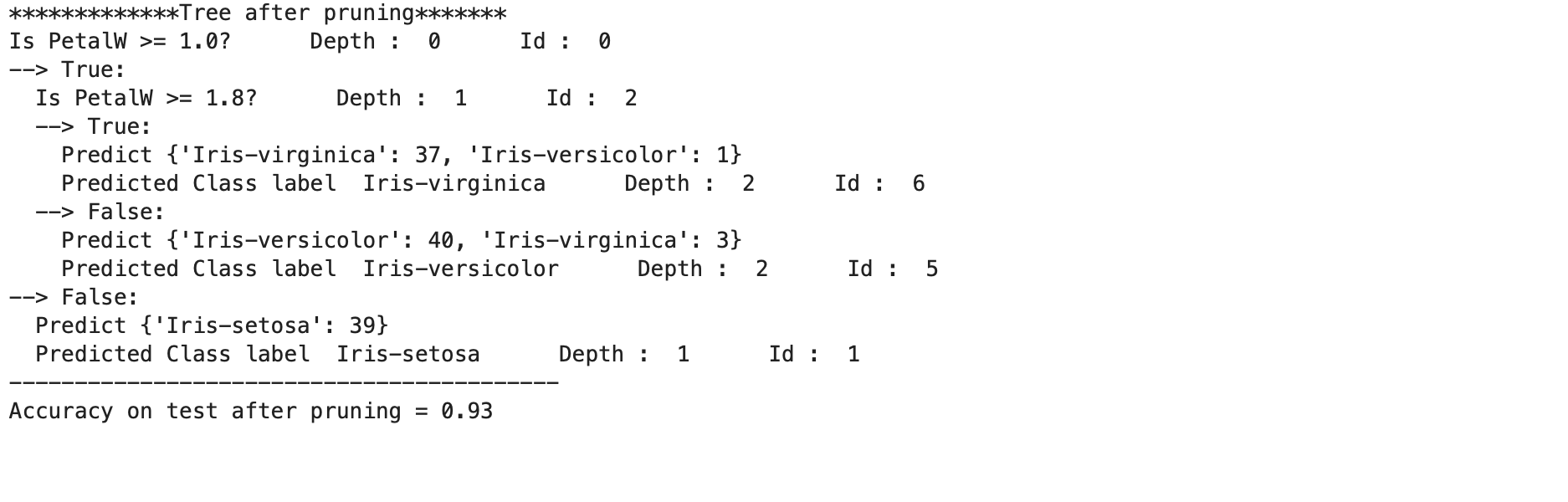
[www.stackoverflow.com](http://www.stackoverflow.com)

[www.facultycs.tamu.edu](http://www.facultycs.tamu.edu)

The Biases of Decision Tree Pruning Strategies: By Tpio Elomaa ( Semantic Scholar)

OUTPUT FOR DATASET BEFORE PRUNING:

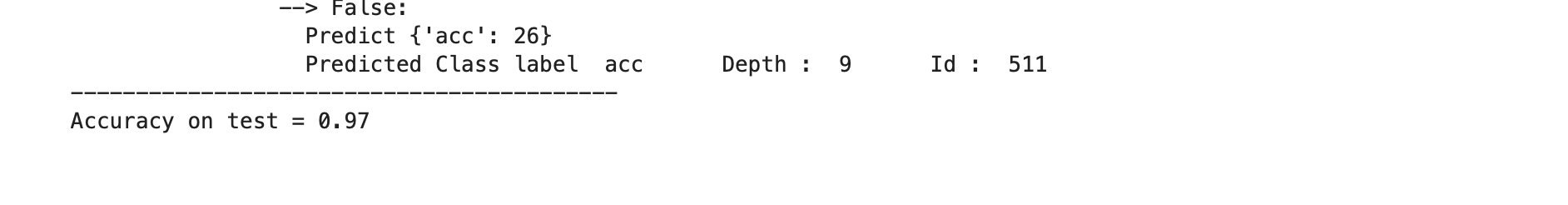
Dataset 1 : Iris.Data

OUTPUT FOR DATASET AFTER PRUNING:

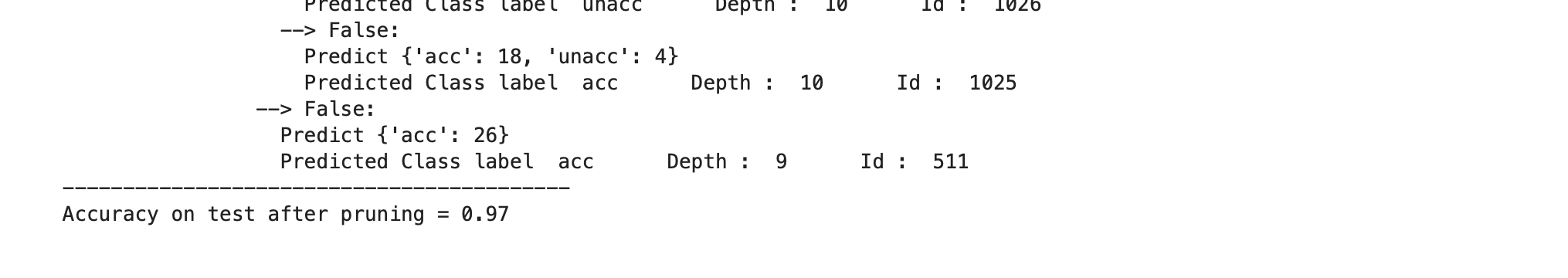
OUTPUT FOR DATASET BEFORE PRUNING:

Dataset 2 : Car.data

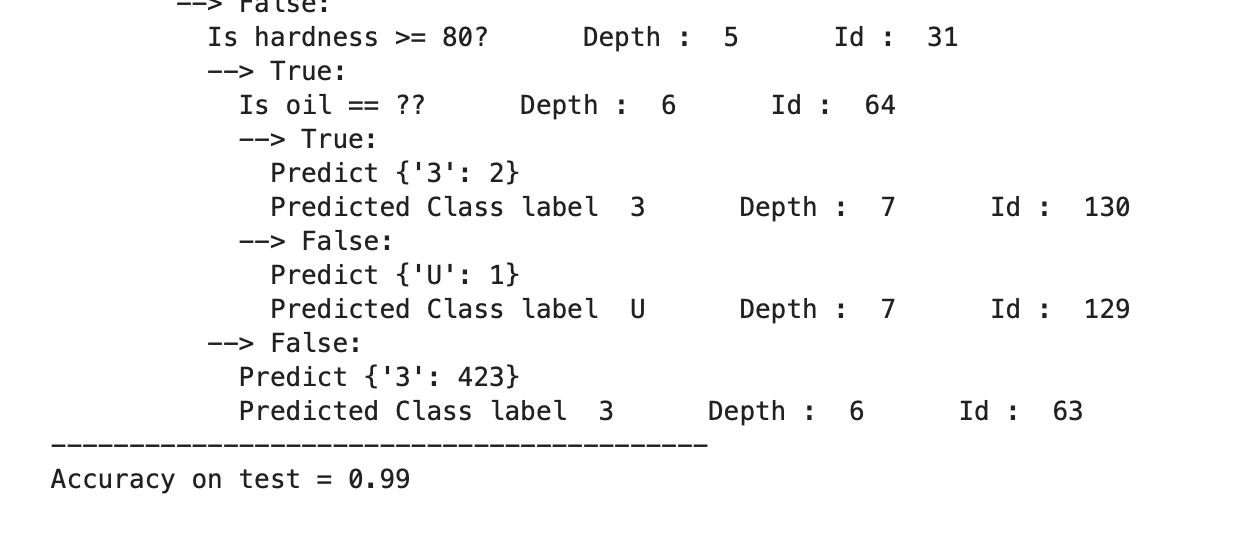
Print\_tree is too big for this data.set to take screenshot. Please check Notebook.ipnyb file in needed.



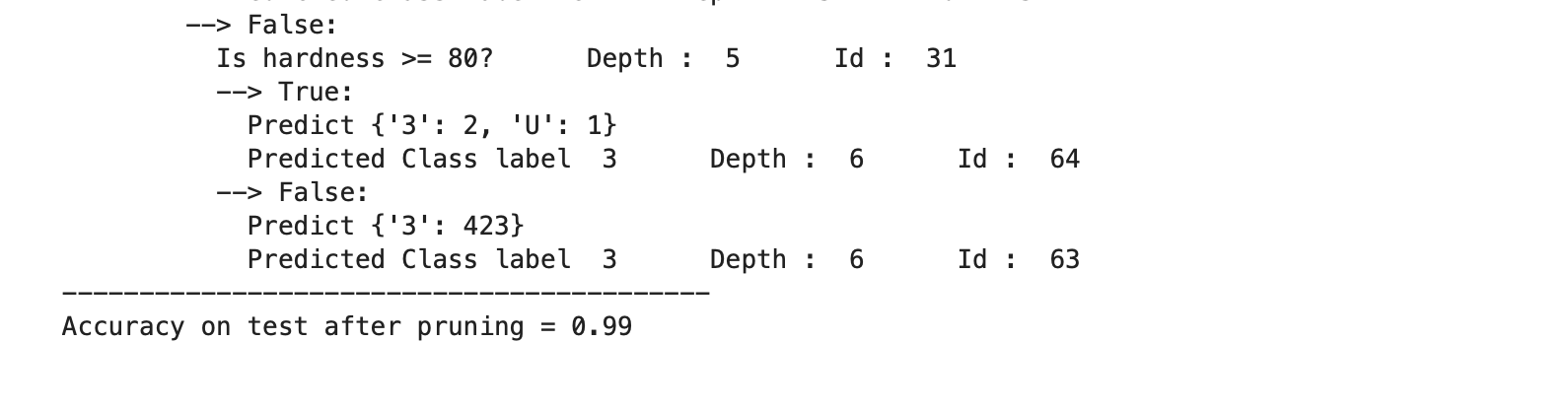
OUTPUT FOR DATASET AFTER PRUNING:



OUTPUT FOR DATASET BEFORE PRUNING:

Dataset 3 : Annealing Data Set

OUTPUT FOR DATASET AFTER PRUNING:



\*\* Note: The header and df are commented for dataset Car.data and Iris.data. Please uncomment and execute to check the results.

Pruning Strategy Used:

Pruning is the technique involved in designing of decision trees, where we can reduce overfitting, by reducing the number of nodes(leaf or internal) , as long as the accuracy of the decision tree remains intact. For our decision algorithm, we have employed Post-Pruning by first sorting the list of inner nodes in the descending order, to identify the highest valued internal nodes. The strategy involves pruning all the internal nodes , starting from the highest valued one, as long as the accuracy remains higher or equal to the accuracy before pruning. This strategy thereby helps to reduce the number of internal nodes as well as their corresponding leaf nodes, while maintaining the accuracy.

The steps that we used to implement pruning is as follows-

Step1: Identify the leaf nodes and the internal nodes of our decision tree by calling the methods getLeafNode() and getInnerNode() respectively. Also, store the accuracy of the decision tree(before pruning).

Step2: Sort the internal nodes in the descending order, in order to identify the nodes with highest ‘Id’ values.

Step3:

1. Start pruning the nodes iteratively from the sorted internal node list, starting from the highest Id valued node.
2. Check the accuracy of the resultant decision tree (after pruning) at each iteration.
3. At each iteration, compare the corresponding accuracy value with accuracy off the tree before pruning. If the accuracy has increased or remains the same, append that internal node value to the list of nodes to be pruned. Whenever the accuracy is found to have reduced, stop the iteration at that point.
4. The resultant ‘pruning\_list’ will contain the list of nodes that can be pruned to ensure that the accuracy is not affected.
5. Finally, prune the decision tree with the nodes in the ‘pruning\_list’ to get an optimized decision tree.