Week 3 - Decision Tree Classifier

MACHINE INTELLIGENCE LABORATORY

Teaching Assistants
vighneshkamath43@gmail.com
sarasharish2000@gmail.com

Decision Trees are one of the easiest and popular classification algorithms to understand and interpret. The goal of using a Decision Tree is to create a training model that can be used to predict the class or value of the target variable by learning simple decision rules inferred from prior data.

The primary challenge in the decision tree implementation is to identify which attributes to consider as the root node at each level. Handling this is known as attribute selection. The ID3 algorithm builds decision trees using a top-down greedy search approach through the space of possible branches with no backtracking. It always makes the choice that seems to be the best at that moment.
Attribute selection in the ID3 algorithm involves various steps such as computing entropy, information gain and selecting the most appropriate attribute as the root node.

In this assignment you are are required to prepare a module that will help any machine learning fresher to use to calculate these heuristic on any **categorical attributed** data

Your task is to complete the code for the function whose skeleton is predefined.

You are provided with the following files:

1. **week3.py**
2. **SampleTest.py**

Note: These sample test cases are just for your reference.

SAMPLE TEST CASE Data

Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Note: This is the same dataset that was used in class for ID3 Algorithm.

Important Points:

1. Please do not make changes to the function definitions that are provided to you. Use the skeleton as it has been given. Also do not make changes to the sample test file provided to you. Run it as it is.
2. You are free to write any helper functions that can be called in any of these predefined functions given to you.
3. Your code will be auto evaluated by our testing script and our dataset and test cases will not be revealed. Please ensure you take care of all edge cases!
4. Do not assume the schema of the hidden test case to be the same as sample test case, although the values will be categorical but the target variable may vary and the number of classes may not be fixed
5. **Point number 4 stresses on the fact that you should not hard code anything. Remember you are designing a module to help for ID3 algorithm of any kind of data.**
6. **The dataset we will be testing against will have N columns with N-1 attributes and Nth column being the target value.**
7. **Ensure you follow convention while returning from these functions.**
8. **The experiment is subjected to zero tolerance for plagiarism. Your code will be tested for plagiarism against every code of all the sections and if found plagiarized both the receiver and provider will get zero marks without any scope for explanation.**
9. **Kindly do not change variables name or other technique to escape from plagiarism, as the plagiarism checker is able to catch such plagiarism.**
10. Hidden test cases will not be revealed post evaluation.

week3 .py

Contains four function

1. get_entropy_of_dataset
2. get_avg_info_of_attribute
3. get_information_gain
4. get_selected_attribute

Function Name	Input Parameter	Return Value
get_entropy_of_dataset	df : pandas dataframe of the given dataset	entropy : Entropy of the entire dataset (int/float)
get_avg_info_of_attribute	df : pandas dataframe of the given dataset attribute: name of the attribute (column name) whose Avg info is to be found Ex: “Temperature”	Avg_info : Average Information of that attribute (int/float)
get_information_gain	df : pandas dataframe of the given dataset attribute: name of the attribute (column name) whose info gain is to be found Ex: “Temperature”	Information_gain : Information gain of that attribute (int/float)
get_selected_attribute	df : pandas dataframe of the given dataset	tuple((information_gains, selected_column) Information_gains: python dictionary with key as column name and value as its information gain Selected column : string,basically the selected column name for split Example: ({'A':0.123,'B':0.768,'C':1.23} , 'C')

1. **You may write your own helper function if needed**
2. **You can import libraries that come built-in with python 3.7**
3. **You cannot change the skeleton of the code**

SampleTest.py

1. This will help you check your code.
2. Note that the test case used in this is same as the graph defined above
3. Passing the cases in this does not ensure full marks ,you will need to take care of edge cases
4. Name your code file as YOUR_SRN.py
5. Run the command **python3 SampleTest.py --SRN YOUR_SRN (incase of any import error use the below command)**

   python3.7 SampleTest.py --SRN YOUR_SRN