LAB 3: Decision Tree Classifier

Decision Trees stand out as one of the most user-friendly and popular classification algorithms, known for their ease of understanding and interpretability. The ultimate goal behind utilizing a Decision Tree is to create a training model that can effectively predict the target variable's class or value. This is achieved by learning simple decision rules derived from historical data.

The crucial part in implementing a Decision Tree lies in determining which attributes to use as the root node at each level—a task referred to as attribute selection. An attribute selection algorithm picks as root the attribute that can most effectively predict the value of the target variable. ID3 an attribute selection algorithm employs a top-down greedy search approach through the potential branches without backtracking. At each step, it chooses the option that seems most promising.

In ID3, attribute selection involves the following steps:

- **Compute Entropy:** Measure dataset disorder to assess impurity before and after attribute-based splits.
- **Calculate Information Gain:** Find the difference between entropies before and after splitting to quantify attribute's contribution to reducing uncertainty.
- **Select Best Attribute:** Choose the attribute with highest information gain as the root node for that level, ensuring accurate predictions**.**

By iteratively applying these steps, ID3 constructs a Decision Tree that learns from training data, predicting target variables for new instances.

About the dataset:

This “Employee Dataset” contains information about employees in a company, including their educational backgrounds, payment tier, gender, and employment-related factors and is used to predict whether an employee will be retained.

Sample dataset

Education	Payment Tier	Gender	EverBenched	LeaverOrNot
Bachelors	2	Male	No	0
Bachelors	0	Female	No	1
Bachelors	2	Female	No	0
Masters	2	Male	No	1
PHD	2	Male	No	0

Note: The following attributes in the dataset have been encoded with their corresponding key:

Education: The educational qualifications of employees.

- **Bachelors**: 0 **Masters**: 1 **PHD:** 2

Payment Tier: Categorization of employees into different salary tiers.

- **Tier 0:** 0 **Tier 1:** 1 **Tier 2:** 2

Gender: Gender identity of employees

- **Male:** 0 **Female:** 1

Ever Benched: Indicates if an employee has ever been temporarily without assigned work.

- **No**: 0 **Yes**: 1

LeaveOrNot: The target variable which is used to indicate if an employee leaves or not

Task:

Complete code for functions whose skeleton has been provided. You are provided with the following files:

1. DecisionTree.py
2. employeeTest.py

DecisionTree.py

This file contains the following functions:

Function Name	Input	Output
get_entropy_of_dataset	tensor: torch.Tensor , tensor representing the given dataset	dataset_entropy: int/float , entropy of the entire dataset
get_avg_info_of_attribute	1. tensor: torch.Tensor , tensor representing the given dataset 2. attribute: int , number representing the attribute	avg_info: int/float , average Information of that attribute
get_information_gain	1. tensor: torch.Tensor , tensor representing the given dataset 2. attribute: int , number representing the attribute	information_gain: int/float , information gain of that attribute
get_selected_attribute	tensor: torch.Tensor, tensor representing the given dataset	Result: tuple(information_gains,selected_attribute) where information_gains: python dictionary with key as attribute number and value as its information gain selected_attribute : int , attribute number of chose attribute

Complete above functions to implement attribute selection

employeeTest.py

1. This will help you check your code.
2. Rename DecisionTree.py file to CAMPUS_SECTION_SRN_Lab3.py
3. Run the command `python3 employeeTest.py --ID CAMPUS_SECTION_SRN_Lab3 --data employeeData.csv`

Important Points

1. Do not make changes to the function definitions that are provided to you. Use the skeleton as it has been given.
2. Do not make changes to the test file provided to you. Run as is.
3. Do not hardcode values. You are designing a module that has helper functions to run the ID3 algorithm for any kind of dataset. The functions must be designed to be independent of the schema of the dataset.
4. In the dataset provided, the last column has the target variable. The previous columns are explanatory variables.
5. You may write additional helper functions.
6. You can use built-in modules.
7. **You must use PyTorch**

Submission Guidelines

You are to submit two files:

1. The python solution: CAMPUS_SECTION_SRN_Lab3.py
2. Screenshot of Test cases (Single screenshot of all test cases in terminal): CAMPUS_SECTION_SRN_Lab3.png (or jpg)

The google form link for submission will be provided

You won't be able to resubmit, so kindly check your files. Remove all print statements. Resubmitting from different mail will lead to zero marks. Failing some hidden cases will lead to partial marks.

Test cases provided to you are for reference only, hidden test cases will be similar