HW1 report DSCI 552(Machine learning)

**Group Members:**  
 Muhammad Rizwan Malik  
 Hamza Belal Kazi

**Work Breakdown:**

File reading and storing in Table data structure: Hamza

Implementation of Decision Tree algorithm methods (Entropy, Info Gain etc.): Both Team members

Data Structure for Storing Decision tree and Printout methods: Rizwan

**Description of Data Structures used and prinout:**

* **Table:** After reading the given dt\_data.txt file the comma separated values are stored in a NumPy 2d array.
* **Decision Tree Data Structure:** Decision tree is a recursive data structure comprised of Node type data structure. Each Node has 2 fields:
  + Feature: The attribute {Occupied, VIP, Location, etc.} of that Node. This is the attribute on which the tree is being split.
  + Children: A list data structure which stores children of this node as tuples (Child\_Node, Feature\_value). Feature\_Value is the type of the parent attribute on which branching occurred and Child\_Node is again a Node type data structure.
* **Decision Tree Method:** The method decision\_tree() makes recursive calls to itself after deciding the best split which is based on the following criterion:
  + Out of all the available attributes, which one gives the maximum information gain?
* After deciding on the best attribute to split, a for loop iterates over all the different types of this specific attribute. Inside the for loop a recursive call is made going 1 level deeper in the tree until it reaches one of the following 2 base cases:
  + All the attributes have been exhausted.
  + A sub-database has been reached which is ‘pure’ with regards to the ‘label’ i.e., it either has all Yes or all No in the last column.
* After hitting the base case, the ‘label’ is passed as the ‘feature of the node then this Node is added to the children list of the current node and the loop continues. Upon termination, the method returns the root node of the tree.
* **Printout:** The line with ‘ Node Depth (*Number)*: *Attribute Name*’ represents a node in the tree and at what depth it is starting with root=0. The lines with ‘Case : *Attribute value*’ represent the branches coming out of the node. Indentation helps see which branches are coming from which node as all branches of same node have same indentation level. The leaf nodes printout is special, instead of attribute name that have Yes/No showing result(Enjoyment) predicted value.
* Following is the graphical representation of the decision tree printed by our code. It was hand traced based on the printout.

A close up of a map

Description automatically generated

**Requested Prediction:**

(occupied = Moderate; price = Cheap; music = Loud; location = City-Center;

VIP = No; favorite beer = No).  
According to our decision tree the prediction for this case is **Enjoyment=YES**