Programming Assignment 1: Decision Trees

Kavya Sethuram(ksetura@usc.edu); Rasika Guru (rguru@usc.edu); Roshani Mangalore (rmangalo@usc.edu)

1. Implementation of ID3 Algorithm in Python

A. DataStructures used in the Implementation are:

- ***Tree***: A tree is a datastructure that consists of nodes and its connections are called edges. The bottom nodes are also named leaf nodes. We have implemented trees using classes in Python.
- ***Dataframe***: DataFrame is a 2-dimensional labeled data structure and is the most commonly used panda object. We have generated Dataframe from the datafile using pandas.
- ***Dictionary:*** Dictionaries in python are unordered data structures that are used to store unique keys and their corresponding values.

B. Explanation of Modules:

Module 1: Computation of Entropy:

Entropy is calculated using the values of the class label for the Data set.

The formula used is as below

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Module 2: Computation of Information Gain

Information gain is calculated for all the features using the below formula

$$G(S, A) = I(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} I(S_v)$$

The Feature with highest Information Gain is chosen as a parent node for subsequent branches.

Module 3: Building the Decision tree

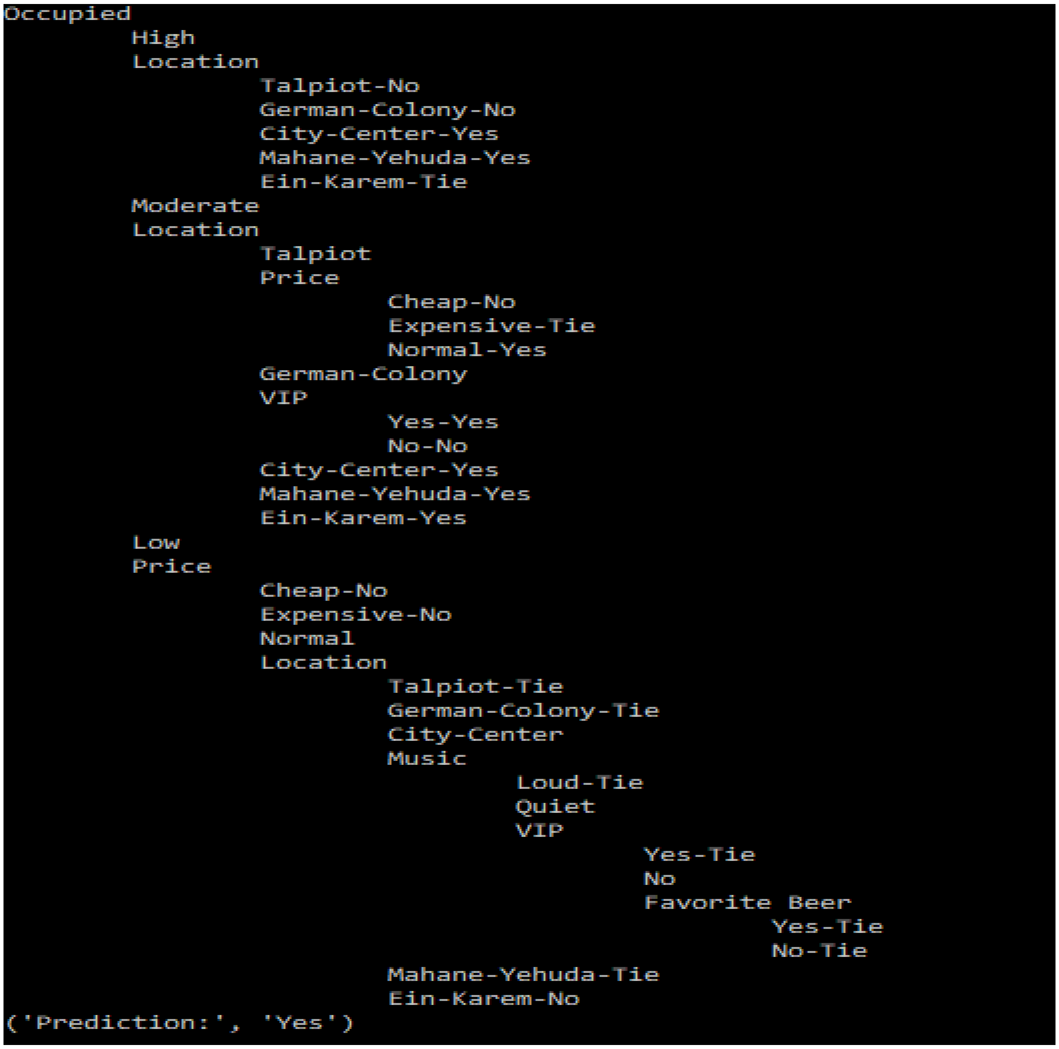
This function is used to build the decision tree. It also includes

- Handling of terminating conditions
- Break ties between the attributes having same gain. The attribute occurring first in the attribute list is given the first preference.
- Calculates the most common labels in a list of labels. It is used to find the majority class (yes/no) in the data after we run out of attributes

Module 4: Printing the Decision tree

This function prints the decision tree in the below format (Screenshot of the output is as below)

- The attribute on the first level is the attribute with the most Information Gain.
- Second level is one of the unique class values of the parent attribute.
- Third level is an attribute with high information gain for the inner branch.
- Further levels will have other attribute values terminating with a class label.



```
Occupied
        High
        Location
                Talpiot-No
                German-Colony-No
                City-Center-Yes
                Mahane-Yehuda-Yes
                Ein-Karem-Tie
        Moderate
        Location
                Talpiot
                Price
                        Cheap-No
                        Expensive-Tie
                        Normal-Yes
                German-Colony
                VIP
                        Yes-Yes
                        No-No
                City-Center-Yes
                Mahane-Yehuda-Yes
                Ein-Karem-Yes
        Low
        Price
                Cheap-No
                Expensive-No
                Normal
                Location
                        Talpiot-Tie
                        German-Colony-Tie
                        City-Center
                        Music
                                Loud-Tie
                                Quiet
                                VIP
                                        Yes-Tie
                                        No
                                        Favorite Beer
                                                Yes-Tie
                                                No-Tie
                        Mahane-Yehuda-Tie
                        Ein-Karem-No
('Prediction:', 'Yes')
```

Module 5: Prediction

This function is used to predict the class label of the following test Instance

```
{'Occupied': 'Moderate', 'Price': 'Cheap', 'Music': 'Loud', 'Location': 'City-Center',
'VIP': 'No', 'FavouriteBeer': 'No'}
```

C. Code Level Optimizations:

- Using list comprehensions over explicit for loops in certain places, making code more compact and execution faster.
- Modularizing parts of code into methods, making code more readable.
- Used Set data structure instead of list to maintain remaining attributes, Set in python is inherently faster than list operations.

D. Challenges:

- The code was initially written in python 3.5, and later wouldn't work in python 2.7, which required debugging. Integer division was dropping decimal values in entropy/gain calculation. Had to explicitly force numerator or denominator to be float before division in 2.7, where as 3.5 took care of it by default.
- Deciding the data structure to store the tree was a challenge. Decided to write our own class to store tree node.

2. Software Familiarization:

We have used the following packages to implement Decision Tree

NumPy

- NumPy is a Numeric Python module. It provides fast mathematical functions.
- Numpy provides robust data structures for efficient computation of multi-dimensional arrays & matrices.
- We used numpy to read data files into numpy arrays and data manipulation.

Pandas

- Provides DataFrame Object for data manipulation
- Provides reading & writing data b/w different files.
- DataFrames can hold different types data of multidimensional arrays.

Existing Libraries:

1. Scikit-Learn (for Python)
   - It's a machine learning library. It includes various machine learning algorithms.
   - We are using its train_test_split, DecisionTreeClassifier, accuracy_score algorithms. This Library uses C4.5 algorithm to Implement Decision tree in Python

2. data.tree (for R)
   - data.tree is the most used library to implement ID3 Algorithm in R
   - This package provides methods for printing and plotting trees. It supports conversion from and to data.frames, lists, and other tree structures from the ape package, igraph, and other packages.
   - These structures are bi-directional and ordered. It helps to navigate from parent to children and vice versa. Ordered means that the sort order of the children of a parent node is well-defined.

3. Applications of Decision Tree

a. Student Selection Model

Student Admission is usually done by comparing candidate application file, so the subjectivity of assessment is most likely to happen because of the lack standard criteria that can differentiate the quality of students from one another. By applying data mining techniques classification, we can build a model selection for new students which includes criteria to certain standards such as the area of origin, the status of the school, the average value and so on. These criteria are determined by using rules that appear based on the classification of the academic achievement (GPA) of the students in previous years who entered the university through the same way. The decision tree method with C4.5 algorithm can be used here. The results show that students are given priority for admission is that meet the following criteria: come from a specific geographic area, public school, majoring in science, an average value above 75, and have at least one achievement during their study in high school

b. Predicting Library Book Use

Forecasting book usage helps librarians to select low-usage titles and move them to distant and less expensive off-site locations that use compact and lesser
storage techniques. For this task, it is important to adopt a book choice strategy that minimizes the expected frequency of requesting removed titles. For any choice policy, this frequency depends on the percentage of titles that have be removed for off-site storage (as measured by the capacity of the main library);
the higher this percentage is, the higher this frequency is expected to be. Decision Trees are used to predict the usage of books in a library.

c. Healthcare and Sensitivity Analysis

Decision tree analysis in healthcare can be applied when choices or outcomes of treatment are uncertain, and when such choices and outcomes are significant (wellness, sickness, or death). The idea of assigning values to states of health may range from a score of 1 for perfect health, 0 for death, and somewhere in between for sickness to a number.This approach allows physicians to better identify the most favorable option for patients. With the range of modeling techniques available, it can also yield valuable information and risk to patients might be reduced.

4. Individual Contribution

Entropy and Information Gain	Kavya Sethuram and Rasika Guru
Decision Trees and handling common labels and ties	Rasika Guru and Roshani Mangalore
Main function (Reading the file, handling terminating conditions), Printing the Decision Tree and Predicting the label for Test data	Roshani Mangalore and Kavya Sethuram