**Department of Computing**

**CS370: Artificial Intelligence**

**Class: BSCS-10AB**

**Lab 10: Identification Trees**

**Date: 20-04-2023**

**Instructor: Dr. Hashir Kiani**

**Lab Engineer: Ms Shakeela Bibi**

# Lab 10: Identification Trees

**Introduction:**

An Identification Tree is a representation of a function that maps a vector of attribute values to a single output value—an “Identification”. An identification tree reaches its decision by performing a sequence of tests, starting at the root, and following the appropriate branch until a leaf is reached. Each internal node in the tree corresponds to a test of the value of one of the input attributes, the branches from the node are labeled with the possible values of the attribute, and the leaf nodes specify what value is to be returned by the function.

**Lab Task:**

Train an Identification Tree classifier considering the following learning problem: the problem of deciding whether to wait for a table at a restaurant. For this problem the output, is a Boolean variable that we will call WillWait; it is true for examples where we do wait for a table. The input,, is a vector of ten attribute values, each of which has discrete values:

1. ALTERNATE: whether there is a suitable alternative restaurant nearby.

2. BAR: whether the restaurant has a comfortable bar area to wait in.

3. FRI/SAT: true on Fridays and Saturdays.

4. HUNGRY: whether we are hungry right now.

5. PATRONS: how many people are in the restaurant (values are None, Some, and Full).

6. PRICE: the restaurant’s price range ($, $$, $$$).

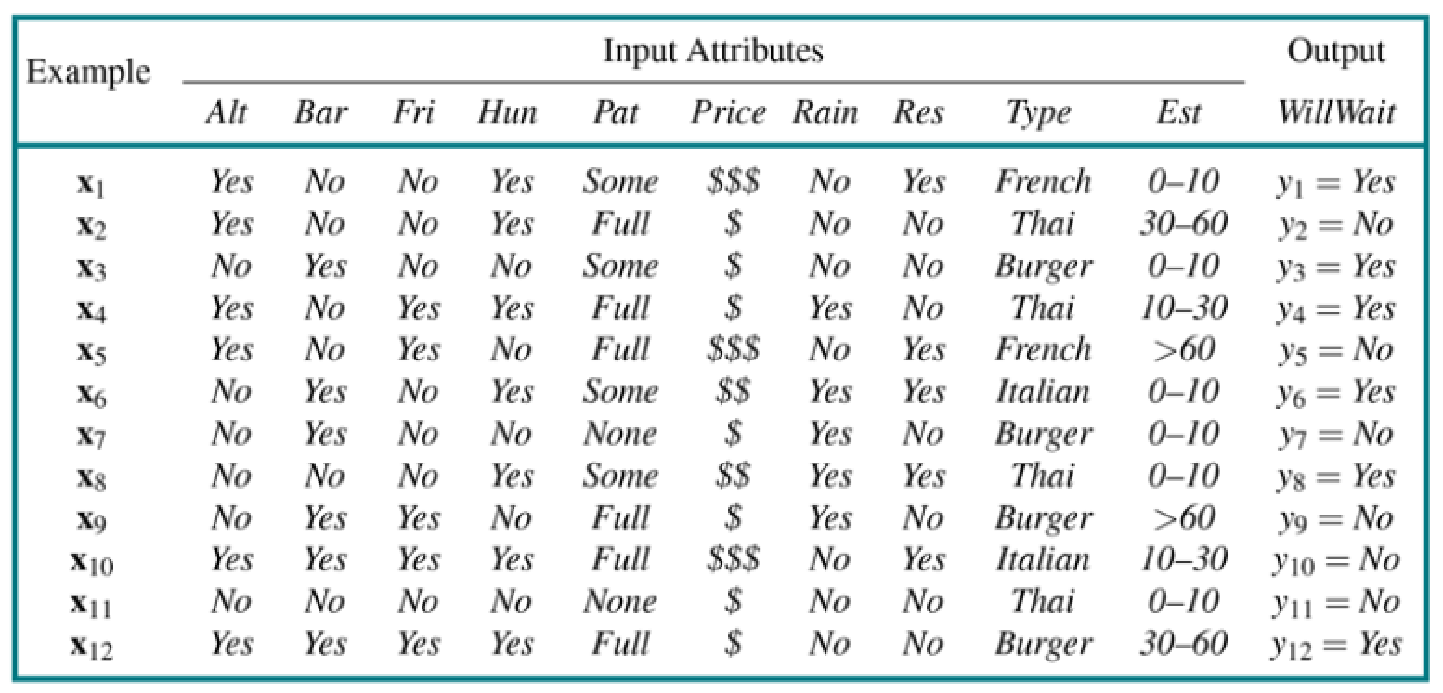
7. RAINING: whether it is raining outside.

8. RESERVATION: whether we made a reservation.

9. TYPE: the kind of restaurant (French, Italian, Thai, or burger).

10. WAITESTIMATE: host’s wait estimate: 0-10, 10-30, 30-60, or >60 minutes.

A set of 12 examples, taken from the experience of one individual, is shown in the Figure at the end. Randomly select 10 examples for the training set and 2 examples for validation set. Then use the training set to build your identification tree. The best solution in this case would give a general function that takes a training set as input and returns the best possible identification tree using the ranking disorder approach discussed in the lecture. Once the tree has been built, test the performance of your tree on the validation set. The best solution in this case would be a general function that takes an identification tree and the validation set as input and returns the number of correct predictions on the validation set.



**Code:**

**import math**

**# Define the attributes and their possible values**

**attributes = {**

**'ALTERNATE': ['Y', 'N'],**

**'BAR': ['Y', 'N'],**

**'FRI/SAT': ['Y', 'N'],**

**'HUNGRY': ['Y', 'N'],**

**'PATRONS': ['None', 'Some', 'Full'],**

**'PRICE': ['$', '$$', '$$$'],**

**'RAINING': ['Y', 'N'],**

**'RESERVATION': ['Y', 'N'],**

**'TYPE': ['french', 'thai', 'burger', 'italian'],**

**'WAITESTIMATE': ['0-10', '10-30', '30-60', '>60']**

**}**

**# Define the training data**

**X\_train = [**

**['Y', 'N', 'N', 'Y', 'SOME', '$$$', 'N', 'Y', 'FRENCH', '0-10'], # Example 1**

**['Y', 'N', 'N', 'Y', 'FULL', '$', 'N', 'N', 'THAI', '30-60'], # Example 2**

**['N', 'Y', 'N', 'N', 'SOME', '$', 'N', 'N', 'BURGER', '0-10'], # Example 3**

**['Y', 'N', 'Y', 'Y', 'FULL', '$', 'Y', 'N', 'THAI', '10-30'], # Example 4**

**['Y', 'N', 'Y', 'N', 'FULL', '$$$', 'N', 'Y', 'FRENCH', '>60'], # Example 5**

**['N', 'Y', 'N', 'Y', 'SOME', '$$', 'Y', 'Y', 'ITALIAN', '0-10'], # Example 6**

**['N', 'Y', 'N', 'N', 'NONE', '$', 'Y', 'N', 'BURGER', '0-10'], # Example 7**

**['N', 'N', 'N', 'Y', 'SOME', '$$', 'Y', 'Y', 'THAI', '0-10'], # Example 8**

**['N', 'Y', 'Y', 'N', 'FULL', '$', 'Y', 'N', 'THAI', '>60'], # Example 9**

**['Y', 'Y', 'Y', 'Y', 'FULL', '$$$', 'N',**

**'Y', 'ITALIAN', '10-30'] # Example 10**

**]**

**# Define the target variable for the training data**

**y\_train = ['Y', 'N', 'Y', 'Y', 'N', 'Y', 'N', 'Y', 'N', 'N', 'Y']**

**# Define the validation data**

**X\_val = [**

**['N', 'N', 'N', 'N', 'NONE', '$', 'N', 'N', 'THAI', '0-10'], # Example 11**

**['Y', 'Y', 'Y', 'Y', 'FULL', '$', 'N', 'N', 'BURGER', '30-60'] # Example 12**

**]**

**# Define a function to calculate the entropy of a set of labels**

**def entropy(labels):**

**unique\_labels = set(labels)**

**entropy = 0**

**for label in unique\_labels:**

**prob = labels.count(label) / len(labels)**

**entropy -= prob \* math.log(prob, 2)**

**return entropy**

**def predict(X\_val, tree):**

**predictions = []**

**for example in X\_val:**

**node = tree**

**while isinstance(node, dict):**

**attribute = node['attribute']**

**value = example[attributes[attribute].index(node['value'])]**

**node = node['children'][value]**

**predictions.append(node)**

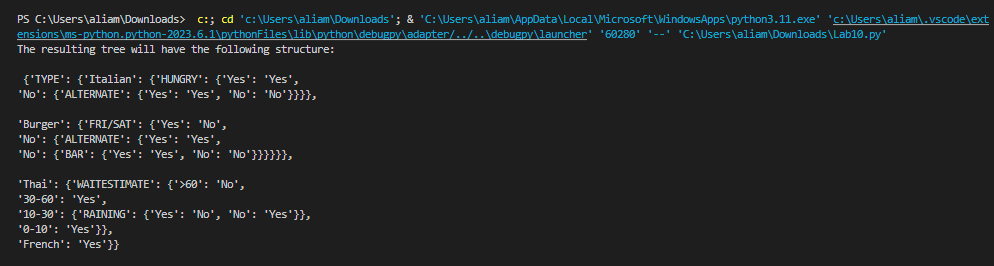
**return predictions**

**tree = build\_tree(X\_train, y\_train, attributes)**

**predictions = predict(X\_val, tree)**

**print(predictions)**

**Screenshot:**

****

**Findings:**

**This identification tree can be interpreted as follows:**

* If the TYPE of the restaurant is Italian and we are hungry, then we will wait if there is an alternative restaurant nearby; otherwise, we will not wait.
* If the TYPE of the restaurant is Burger and it is not Friday or Saturday, then we will wait if there is an alternative restaurant nearby or if the restaurant has a comfortable bar area to wait in; otherwise, we will not wait.
* If the TYPE of the restaurant is Thai and the host's wait estimate is 30-60 minutes, then we will wait if it is not raining outside; otherwise, we will not wait.
* If the wait estimate is less than or equal to 10 minutes or between 10-30 minutes, then we will wait regardless of the weather.
* If the wait estimate is greater than 60 minutes, then we will not wait. If the TYPE of the restaurant is French, then we will wait regardless of the other attributes.