**Department of Software Engineering**

**CS471: Machine Learning**

**Class: BESE-12**

**Lab 5: Decision Trees**

**Date: 08-03-2024**

**Instructor: Dr. Hashir Kiani and Dr Duad Abdullah**

**Lab Engineer: Ms Iram Tariq Bhatti**

# Lab 5: Decision Trees

**Introduction:**

A Decision Tree is a representation of a function that maps a vector of attribute values to a single output value—a “decision”. A decision 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.

**Objective:**

* Understand and implement decision trees.
* Learn encoding techniques.
* Using graphviz to print the decision tree.

**Tools:**

Google colab or equivalent environment

**Lab Task:**

Train a Decision 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 decision tree. The best solution in this case would give a general function that takes a training set as input and returns the best possible decision tree using the information gain 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 a decision tree and the validation set as input and returns the number of correct predictions on the validation set. You are required to print decision tree at the end as well.



Note: You need to code your own decision tree algorithm for this task. Do not use any functions from ML libraries like ScikitLearn.

**Deliverable:**

Submit jupyter notebook with code and outputs before deadline on LMS.

**Refences:**

1. Encoding techniques: <https://medium.com/anolytics/all-you-need-to-know-about-encoding-techniques-b3a0af68338b>