Readme for Project 1

Submitted by Abhijay Mitra (18IE10001)

Instructions to run the file

- The code is submitted as a single python3 file `18IE10001_p1.py` which can be run on any machine with the libraries installed with the command:
  `python3 18IE10001_p1.py`.
- The libraries used in the program are math, json and pandas and have been solely used for the purpose of import and extraction of data, and printing data in an orderly fashion. All necessary functions and methods for the purpose of the project have been self generated.
- In case of difficulty to run the code with the github links for extracting the training and test data (lines 153 and 158) due to internet or server issues, feel free to use any offline copy of the same.

Detailed explanations to the major functions/methods in the code

- **`attribute_for_optimal_splitting_at_node`**: This methods is called on a dataset corresponding to a particular node. It returns the attribute which will be responsible at the split in the current node. The attribute to be returned is found using the information gain of the attributes. The one having the maximum information gain is the attribute that shall be returned.
- **`majority_finding_terminating_case`**: This is a terminating case to the recursion we go through while building the decision tree. We arrive at this case only if there are are more attributes left and hence the majority class is used to decide the value in the node.
- **`single_class_terminating_case`**: This is another terminating case to the recursion we go through while building the decision tree. If all values of an attribute belong to 1 class, we arrive at this base case. Here we create the leaf node with a value of that class.
- **`tree_generator_from_training_set`**: This is the recursive code forming the heart of the algorithm. It generates the decision tree from the training data. It takes as input the dataset corresponding to a particular node.
  It calls the method **`attribute_for_optimal_splitting_at_node`** to find the attribute to be used for splitting. Then it splits the node and recursively calls itself on its children. Finally it returns the subtree formed at the particular node and hence forms the entire decision tree recursively in the top down fashion.
- **`predict_using_test_set`**: This function returns a tuple containing the prediction and the accuracy given the test data set. It uses the decision tree for the prediction.

Contact

In case of any further queries feel free to contact

- Abhijay Mitra
- mitraabhijay@gmail.com