README

Brief Description:

The **Analysis** directory contains the following files and folders:

- **Variable Depth Binary Decision Tree Approach** : Variable Depth Binary Decision Tree Approach is a folder which has two folders as follows.
  - **gini** : The gini folder contains the image of the decision tree before pruning named as **decison_tree_before_pruning.pdf** and the pruned decision tree named as **decison_tree_after_pruning.pdf** formed using gini as the impurity measure and keeping reuse_atttribute value to True. It also contains graphs for depth, nodal and accuracy analysis.
  - **entropy** : The entropy folder contains the image of the decision tree before pruning named as **decison_tree_before_pruning.pdf** and the pruned decision tree named as **decison_tree_after_pruning.pdf** formed using entropy as the impurity measure and keeping reuse_atttribute value to True. It also contains graphs for depth, nodal and accuracy analysis.
- **Scatter Plots**: Scatter Plots is a folder which contains the graphs for visualization of the data. It points out important information such as outliers, etc.
- **Report.pdf**: Contains a detailed analysis of the working of Decision Tree Classifier.

The **Project** directory contains the following files and folders:

- **gini** : gini is a folder which gets populated when analysis.py is run and metric gini is used.
- **entropy** : entropy is a folder which gets populated when analysis.py is run and metric entropy is used.
- **Scatter_Plots**: Scatter_Plots is a folder which gets populated if the code corresponding to visualization of data in analysis.py is run. (Visualization Code has been commented out to prevent it from re-runnning)
- **avila_combined.txt** : Contains the data used for training the Decision Tree Classifier. The data is comprised of 20867 samples with 10 continuous valued attributes and one categorical target value.
- **requirements.txt**: Contains all the necessary dependencies and their versions
- **utility.py**: Contains all the helper functions such as gini, entropy etc. used by the above files (if any)
- **models.py** : Contains the implementation of the node class and the DecisionTreeClassifier class. These two classes work together to build the Decision Tree.
- **analysis.py**: Contains the python code we implemented to perform analysis on the Data and Decision Tree Classifier Model.

Directions to use the code

---

1. Download the **Project** directory in into your local machine.
2. Ensure all the necessary dependencies with required version and latest version of Python3 are available (verify with requirements.txt)

```
pip3 install -r requirements.txt
```

3. Run the analysis.py file.
    - It will prompt you to enter a metric to use as impurity measure. Please enter either 'gini' or 'entropy' in all small caps.
    - The next prompt will query you whether or not you want to re-use attributes at different levels of the tree. Please enter either 'True' or False'.
    - The last prompt will ask you to enter the maximum depth upto which you want to grow the Decision Tree. Please enter an integer.
      Important Remarks
    - Note that avila_combined.txt has over 20867 samples. Our code uses 60% of the samples for training, 20% for validation while pruning, and 20% for testing. If you continue with the same, the code might take over two hours to reach completion if the maximum depth specified by you is significantly high.
    - We suggest that you change this split of 60-20-20 to **15-5-80** just for quick execution, verification and validation of the code, since we (Hritaban and Neha) have already analysed and reported our findings with the correct split.
    - The above change can be made easily in by tweaking a value in the analysis.py file. Go to line 94(may vary slightly) in analysis.py and change test_size from 0.2 to 0.8 in the following line of code.

      ```
      X_sub, X_test, y_sub, y_test = train_test_split(X, y, test_size=0.8, random_state=i + 1)
      ```

Final Remarks

The dataset is taken from the following link.
https://archive.ics.uci.edu/ml/datasets/Avila

It contained two different txt files namely :

- avila-tr.txt - a training set containing 10430 samples
- avila-ts.txt - a test set containing 10437 samples

We were asked to combine the two files and then perform the 80/20 split and thus we created a new txt file named avila_combined.txt containing 20867 samples to perform the analysis.