

Command to run the python code

```
1 python dtrees.py [name_of_train_data_file] [name_of_validation_data_file]
   [name_of_test_data_file] [type_of_tree_to_use] [impurity_measure] [
   pruning]
```

For example to run dtree algorithm with entropy and no pruning on the c300_d100 dataset use the given code.

```
1 python dtrees.py train_c300_d100.csv valid_c300_d100.csv test_c300_d100.
   csv -td -ie -pn
```

Now let us elaborate on the command line parameters for the code:-

- **name_of_train_data_file**

Provide the name of training data file with extension (please provide .csv files only)

- **name_of_validation_data_file**

Provide the name of validation data file with extension (please provide .csv files only)

- **name_of_test_data_file**

Provide the name of testing data file with extension (please provide .csv files only)

(All the above files should be in the same folder as the code.)

Please use all these functionalities(mentioned below) as in or the code will give an error of wrong command line arguments.

- **type_of_tree_to_use**

-td for dtrees

- **impurity_measure**

-ie for entropy

-iv for variance

- **pruning**

-pn For no pruning

-pd For depth based pruning

-pr For reduced error pruning

If you are using the random forests then the code doesn't require you to give the pruning and impurity values.

These are the modules/imports required for the code:-

For the decision trees code(all 6):-

```
1 import numpy as np
2 import pandas as pd
3 import os
4 import subprocess
5 import sys
6 import copy as copy
```

These are the modules that need to be installed on the devices on which the code is run. Please install all these before running the code.

The code was made by following the given algorithm:-

ID3(*Examples*, *Target_attribute*, *Attributes*)

Examples are the training examples. *Target_attribute* is the attribute whose value is to be predicted by the tree. *Attributes* is a list of other attributes that may be tested by the learned decision tree. Returns a decision tree that correctly classifies the given *Examples*.

- Create a *Root* node for the tree
- If all *Examples* are positive, Return the single-node tree *Root*, with label = +
- If all *Examples* are negative, Return the single-node tree *Root*, with label = -
- If *Attributes* is empty, Return the single-node tree *Root*, with label = most common value of *Target_attribute* in *Examples*
- Otherwise Begin
 - $A \leftarrow$ the attribute from *Attributes* that best* classifies *Examples*
 - The decision attribute for *Root* $\leftarrow A$
 - For each possible value, v_i , of A ,
 - Add a new tree branch below *Root*, corresponding to the test $A = v_i$
 - Let $Examples_{v_i}$ be the subset of *Examples* that have value v_i for A
 - If $Examples_{v_i}$ is empty
 - Then below this new branch add a leaf node with label = most common value of *Target_attribute* in *Examples*
 - Else below this new branch add the subtree
 $ID3(Examples_{v_i}, Target_attribute, Attributes - \{A\})$
- End
- Return *Root*