

Guidelines to use the Interactive Decision Trees Graphical User Interface (GUI).

1. Import file GUI

In Figure 1 the GUI for importing csv files is shown.

The GUI for importing CSV files consists of the following elements:

- File:** A text input field with the placeholder "Type the file name".
- Random Sampling:** A checkbox.
- Sample Size:** A numeric input field with a value of 0 and up/down arrows.
- Header:** A dropdown menu with the value 0.
- Column:** A dropdown menu with the value 0.
- Delimiter:** A text input field with a comma (,) as the delimiter.
- Split in train-test sets:** A checkbox.
- Test Size:** A numeric input field with a value of 0.25 and up/down arrows.
- Random State seed:** A numeric input field with a value of 0 and up/down arrows.
- Import file:** A button at the bottom.

Figure 1 GUI for importing csv files.

How to use this GUI?

1. Type the name of the file (e.g. filename.csv) in the corresponding 'File' text area widget.
Note: The dataset must live in the working directory.
2. If users want to import only a random sample of the dataset, they should check the 'Random Sampling' widget option and specify the size of the sample (e.g. 1000) in the corresponding 'Sample Size' text area widget. Otherwise leave both widgets blank.
Note: The sample size must be an integer.
3. Specify the which row to be used as Header, which column to be used as row indexes, and the delimiter.

Note: Python starts the numbering from 0 not from 1. E.g.

0	1	2	...
1			
2			
...			

4. If users want to split the dataset into train and test sets check the 'Split in train-test sets' and specify the desired 'Test size' in the text area widget (e.g. 0.25 -> 25% of the dataset will be used for testing and the rest 75% for training)
5. In the 'Random State seed' widget the users can specify an integer to be used as a seed for the random sampling (when splitting in training and test sets using the same seed will ensure that the results will be the same).
6. Click the 'Import file' button.
Once the button is clicked it may take few seconds or minutes (depending on the dataset size) to load the dataset. The GUI does not give an output. If something is wrong an error will be raised.

2. Pre-processing Stage GUI

The first tab is named "Pre-processing Stage" and contains widgets (e.g. text boxes, dropdown menus etc) for:

- a. Defining classes,
- b. Grouping the available features (and assign colours to the groups) and
- c. Selecting important features.

The GUI is shown in Figure 2:

The GUI consists of several sections:

- Class Definition:** A text input for 'Class Name' (placeholder: 'Type the class name'), an 'Add Class Label' button, a 'Pick a color' input (placeholder: 'blue'), a color picker, and an 'Assign color to the Class' button.
- Group Definition:** A text input for 'Group' (placeholder: 'Type the group name'), a 'Features' input (placeholder: 'Assign features to group'), a 'Pick a color' input (placeholder: 'blue'), a color picker, and an 'Assign Color to Group' button.
- Feature Selection:** A list of features on the right: 'h_veg', 'r_st', 'LAI_min', 'LAI_max', and 'Vr'. Below the list is a 'Random Features' checkbox and a 'Total Features' counter showing '0'. A 'Select Features' button is at the bottom right.

Figure 2 Pre-processing Stage GUI

2.1 Define Classes

In Figure 3 the interface for defining classes is shown.

This section focuses on the class definition part of the interface, showing the text input for the class name, the button to add the label, the color selection process (input field and picker), and the button to assign the color to the class.

Figure 3 User Interface for Defining classes

How to use it?

1. Type the name of the class in the 'Class Name' text area widget.
Note: The names of the classes should be the same as the ones included in the csv file we imported.
2. Click the 'Add Class Label' button
3. Click on the coloured box to open the colour picker box. Pick a colour and click ok. Or, type the name of the colour in the 'Pick a color' text area widget.
4. Click the 'Assign colour to the Class' button

2.2 Pre-Group Variables and colour code the groups

In Figure 4 the interface for grouping the variables into groups and picking a colour for each group is shown.

This section focuses on the group definition part of the interface, showing the text input for the group name, the button to assign features to the group, the color selection process (input field and picker), and the button to assign the color to the group.

Figure 4 User Interface for grouping the variables into groups and picking a colour.

How to use it?

1. Type the Group name in the 'Group' text area widget (e.g. Geophysical Properties)
2. Type the names of all the variables belonging to that group in the 'Features' text area widget.
Note 1: The variables should be space separated
Note 2: The variables names should be the same as they are in the input file dataset
3. Click the 'Assign Features to Group' button
4. Click on the coloured box to open the colour picker box. Pick a colour for the group and click ok. Or, type the name of the colour in the 'Pick a colour' text area of the widget.
5. Click the 'Assign Colour to Group' button.

2.3 Select Important Features GUI

In Figure 5 the interface for selecting the important variables of the dataset is shown.

Features

- h_veg
- r_st
- LAI_min
- LAI_max
- Vr

☐ Random Features

Total Features 0

Select Features

Figure 5 User Interface for selecting the important variables of the dataset.

How to use it?

1. Select the important variables of the dataset from 'Features' window widget.
Note: To select multiple variables press and hold ctrl and click on the variable name.
2. If users want extra variables to be selected randomly (besides the ones they already have chosen) then the 'Random Features' should be checked.
3. If users have checked the 'Random Features' then they need to specify the total number of variables that eventually the algorithm will use in the 'Total Features' text area widget.

For example, if the user has selected 5 variables as important, has checked the Random Features option and specifies 10 Total Features then 5 extra variables (other than the ones already selected by the user) will be randomly selected.

2.4 Interactive Construction and Analysis of Decision Trees

In Figure 6, 7 and 8 the interface for Interactive construction and analysis of Decision Trees is shown.

Variable Name: Group Name: Equation:

criterion: gini max_depth: 10 max_features: 34 max_leaf_nodes: 18

min_impurity_decrease: 0.00001 min_samples_leaf: 2 min_samples_split: 2 random_state: None

splitter: best nodes coloring: Classes edges_shape: Lines plot_width: 1400

plot_height: 800 marker size: 15 text size: 15 edges opacity: 0.7

nodes opacity: 1 ☒ Best_first_Tree_Builder

Node_id: 0 Features: h_veg Split Point: 0 Max_leaf_nodes_left_subtree: 2

Max_leaf_nodes_right_subtree: 2 ☐ Refresh ☐ Tree is Pruned

Figure 6 Interface for Interactive construction and analysis of Decision Trees is shown

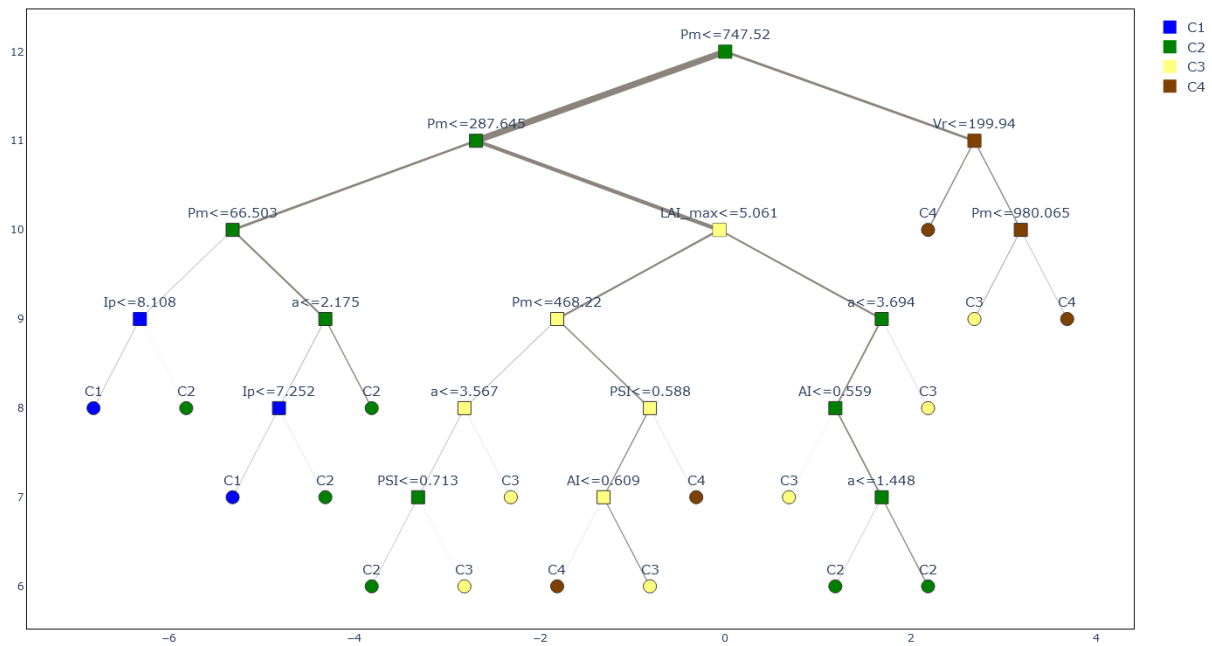


Figure 7 Visualization of DT.

Node to prune

☐ Modified

☐ Refresh

Tree State Leaf Node ID Class

Figure 8 Interface for Manual Pruning and change of leaf node class.

2.4.1 How to create a new composite variable?

In the widgets of Figure 4, the ones used in the creation of a new composite variable are those of 1st line.

1. Type the name of the new variable in the 'Variable Name' widget.
2. Type the group name the new variable should belong in the 'Group Name' text area widget.
3. Type the equation in the 'Equation' text area widget.
4. Click the 'Create Feature' Button
5. Click the 'Update Features' Button.

2.4.2 Tuning Parameters widgets

Lines 2 to 6 in Figure 6 contain widgets which control the tree structure. Whenever, the user changes the value of one of these widgets the plot of the DT is automatically updated.

How each parameter control tree structure?

- Criterion = function to measure the quality of a split. The two options are: "gini" for the Gini impurity and "entropy" for the information gain.
- max_depth = The maximum depth of the tree
- max_features = The number of variables to consider when looking for the best split. For example, if the dataset contains 28 variable and this widget value is fixed to 5 then the algorithm will use 5 variables (randomly selected)
- max_leaf_nodes = Build a tree with max_leaf_nodes.

- `min_impurity_decrease` = a node will be split if this split induces a decrease of the impurity greater than or equal to this value.
- `min_samples_leaf` = The minimum number of data points required to be at a leaf node. A split point at any depth will only be considered if it leaves at least `min_samples_leaf` data points in each of the left and right branches. This may have the effect of smoothing the model.
- `min_samples_split` = The minimum number of data points required to split an internal node.
- `Splitter` = The strategy used to choose the split at each node. Supported strategies are “best” to choose the best split and “random” to choose the best random split.
- `nodes coloring` = The strategy used to colour the nodes. Supported strategies are “Impurity” where the nodes are colored based on their impurity (expressed by gini or entropy index), “Classes” where the nodes are coloured based on the class they belong, “Features_color_groups” where the nodes are coloured based on the color of the group the variable to split of the node belongs to.
- `Edges shape` = The shape of the edges. Supported options are “Lines” for lines and “Lines-steps” for lines in steps.
- `Plot_width`, `Plot_height` = the width and height of the plot, respectively.
- `Marker_size` = the sizes of the squares (nodes) and circles (leaf nodes)
- `Text_size` = the size of the text in the plot.
- `Edges_opacity`, `nodes_opacity` = the edges and nodes opacity respectively. A float number from 0 to 1.

2.4.3 How to Manually change variable and threshold to split in nodes?

How to manually change nodes variables and threshold to split using the GUI?

1. In the widget 'Node_id', specify the id of the node for which changes are to be made. The user can see the nodes IDs by hovering over the nodes in the plot (a window displaying the ID, the impurity and the nodes samples appears).
2. Select the variable to split from the dropdown menu of the widget named 'Features'.
3. Specify the new threshold value in the widget named 'Split Point'
4. Specify the `max_leaf_nodes` the left subtree should have in the widget named 'Max_leaf_nodes_left_subtree'
5. Specify the `max_leaf_nodes` the right subtree should have in the widget named 'Max_leaf_nodes_right_subtree'
6. The “Refresh” widget should be checked when:
 - a. The user wants to undo all the changes made by the user. In this case “Tree is Pruned” widget should be unchecked.
 - b. the user has previously modified the tree, then pruned it and then wants to modify it again. In this case “Tree is Pruned” widget should be checked too.
7. If the tree is previously pruned and needs to be modified then the ‘Tree is pruned’ widget needs to be checked.
8. Press the button widget 'Apply Changes'.

Note_1: Every time we manually change the variable and/or threshold in a node of the DT, the algorithm retrieves the data that correspond to the nodes resulting after the new split. Then, it uses these data as input to fit two trees: one for the data that belong to the node of the left branch (e.g. in node 1: $X \leq X_1$) and one for the data that belong to the node of the right branch (e.g. in node 1: $X > X_1$). So, the two subtrees will be optimal for this local part of the overall DT.

2.4.3 How to Manually prune the DT?

1. Check the 'Tree is Modified' widget if the DT was previously modified (by manually changing nodes variables and threshold values)
2. Check the "Refresh" widget to undo all the prunings made previously:
 - a. If Tree_is_modified is unchecked then it will undo all the prunings, and then for the next prunings it will use as basis the DT resulting from the controlling parameters.
 - b. If Tree_is_modified is checked then it will undo all the prunings, and then for the next prunings it will use as basis the DT resulting from the modifications we made. This is the case where the user has previously modified the tree, then pruned it, modified it again, and then wants to prune it again.
3. Specify node id to be pruned in the 'Node to prune' widget
4. Press the 'Prune' button

2.4.3 How to Manually change leaf node class?

1. Select whether the tree was last modified (e.g. variable or threshold to split was changed) or pruned from the dropdown menu widget named 'Tree State'.
2. Specify the node id (numbering of nodes is based on the last modified or pruned) using the text area widget named 'Leaf Node ID'.
3. Specify the new class of the leaf node using the text area widget named 'Class'
4. Press the 'Change Class' button.

3. Evaluation Metrics and Plots

In Figure 9 the interface for evaluating the DT is shown.

Tree State: No expert tree interactions ▼

☒ Split in train-test sets

Calculate accuracy

Plot Confusion Matrix

File Name: Type the file name

File Format: pickle ▼

Output DT & Data

Figure 9 Interface for the evaluation and export of the DT

How to use it?

1. Specify the tree state using the dropdown widget:
Note: The interaction here refers to interaction with the DT. Creating new variables is an interaction by making changes to the data not the DT itself.
2. Then click on the 'Calculate accuracy' button.
3. Click on the "Plot Confusion Matrix" button.

How to export the tree and training and test datasets?

1. Type the name of the file (e.g. FileName) in the File Name text area widget.
2. Select the file format from the "File Format" dropdown menu widget.
3. Click on the "Output DT & Data" button. This will generate the following output files:
 - FileName: which is a file that stores a table with the DT nodes info.
 - Training_input_FileName: These are the input training data.
 - Training_classes_FileName: These are the input classes data.

- Accuracies_FileName: These are the calculated accuracies.