Note: You can zoom in/out with ‘Z’+‘I’/‘Z’+‘O’.

HOME

Welcome to this information page. Here, we provide a concise explanation of the process involved. You will be required to upload a CSV file, which will serve as the training data. The CSV file can be generated either by using the >_next(log) application or by utilizing a general CSV format. In the latter case, the first $n-1$ columns represent the features, while the n^{th} column corresponds to the labels, target, or classes column.

For your convenience, we have included a settings menu where you can specify the CSV file you wish to parse. Once you have successfully uploaded the CSV file, additional pages and functionalities will become accessible, allowing you to proceed with your desired tasks.

MODEL

On this page, you can browse through different models and make your selection. Once you have chosen a specific model, you will be able to input the corresponding parameters. If you wish to explore a range of parameters to determine the best ones, you can simply add them as a list. For example, if you’re working with the Random Forest model, you can input "n_estimators" as [1, 2, 3, 4]. This will iterate through all the provided parameters and return the optimal one based on the f1 score.

For string inputs, please use the format ["gini", "entropy", "log_loss"], while for boolean values, utilize 0 for False and 1 for True.

Please take note of the additional input fields located at the bottom right of the page, specifically designed for the KNN and LVQ models. These fields serve the purpose of specifying the features you wish to plot on the subsequent page. Here’s how it works:

- If you input only one feature, it will be plotted against itself on the next page.
- When you input two features, three plots will be generated on the next page: feature 1 against feature 1, feature 2 against feature 2, and feature 1 against feature 2.
- Similarly, if you input three features, the same logic applies, and the final plot will be a 3D representation containing all three features.

Please note that if the "use >_next(log)" option is enabled in the settings, certain columns from the dataset will be excluded. Specifically, any features that appear in a rule (after the "then" keyword) will be dropped to prevent 1:1 mappings. Therefore, if you encounter an error indicating that a feature you inputted, which is actually a column in your dataset, has been dropped, it is because it is part of a rule.

You also have the option to leave these fields empty. In such cases, the three most significant features will be automatically extracted using a Gradient Boosting Classifier.

It’s important to note that the choice of the best feature selector depends on the specific dataset. However, modifying the feature selector within the code is a straightforward process, allowing you to experiment with different selectors at your discretion.

At the bottom left of the page, you will find the "Auto-ML" button. Activating Auto-ML allows you to click "Train" and have the program read the contents of the "auto_ml.txt" file. It will then execute the training process on all the models listed within the text file, disregarding any selections made in the GUI. The Results page will display the best model and its corresponding parameters, determined based on the f1 score.

The slider enables you to choose the percentage of the dataset allocated for training/testing.

Please ensure that the format of both the parameter input list within the GUI and the "auto_ml.txt" file strictly adheres to Python syntax. Additionally, remember to use either 0 or 1 for boolean values. For instance, when specifying a list of strings, input ["gini", "entropy"], rather than [gini, entropy]. Furthermore, in "auto-ml.txt" ensure that both the parameter and the value are enclosed in ”. For example: ’criterion’: ’["gini", "entropy"]’

RESULTS

In the model results section, you can examine the outcomes of the (best) model. Within this section, you have the ability to navigate through the utilized parameters and observe various scores. Additionally, depending on the model, you can browse through images, view them in full screen, and save them for future reference.

Furthermore, you have the option to save the model as a .pickle file. This file will retain important information, including the timestamp of when the model was saved, the actual model itself, the percentage of the training split, the currently displayed image in the graphical user interface (GUI), and the f1, precision, and recall scores.

Feel free to explore and utilize these features to analyze and store your model outcomes efficiently.

LOAD & TEST PRETRAINED MODEL

On this page, you have the option to upload a .pickle file containing a saved model. By doing so, you can access valuable information about the model itself. Additionally, you can upload a .csv file to evaluate the model’s performance on a different dataset than the one it was originally trained on. Once the testing is complete, comprehensive statistics pertaining to the training process will be displayed, providing you with a deeper understanding of the model’s behavior and effectiveness.

IMPLEMENTATION

This section offers further details about the implementation. Currently, there are four models available on the page: Decision Tree, Random Forest, GLVQ, and KNN.

On the Results page, for Decision Tree and Random Forest, there are images that illustrate how the model determines the class for each instance.

For GLVQ and KNN, plots will be displayed, where the color of each instance represents the model's prediction.

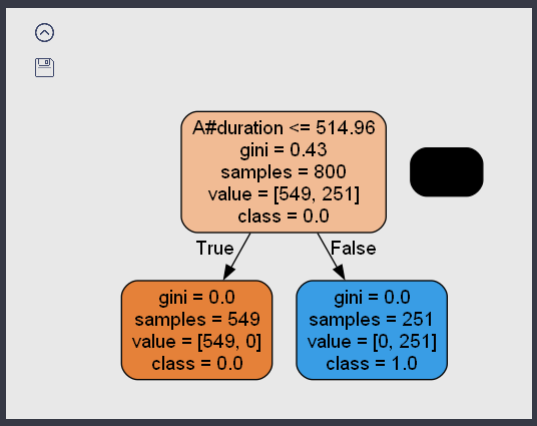

Figure 1: **Decision tree**

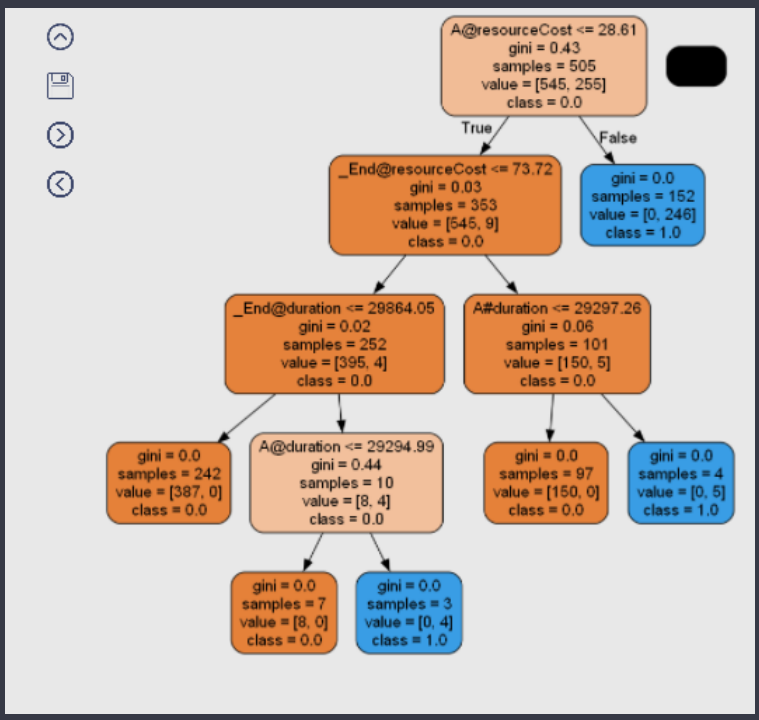

Figure 2: **Random Forest**

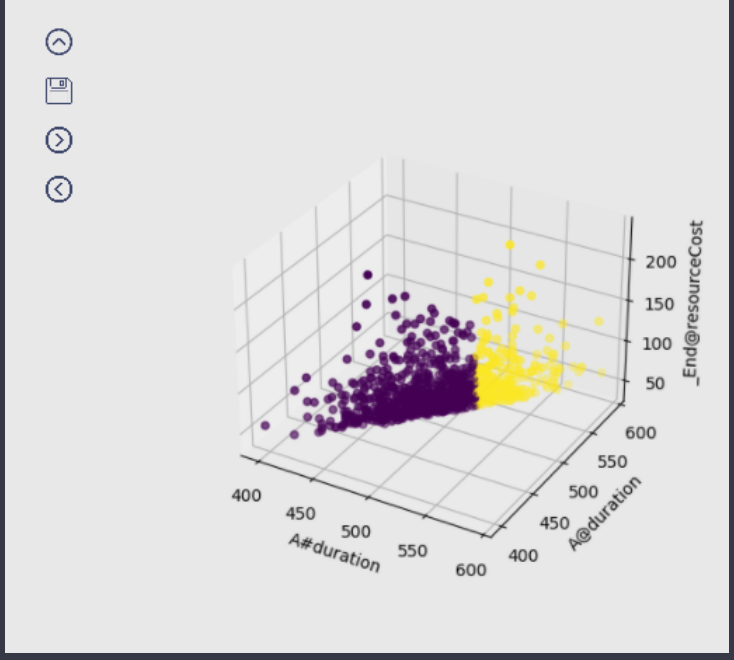

Figure 3: **GLVQ**

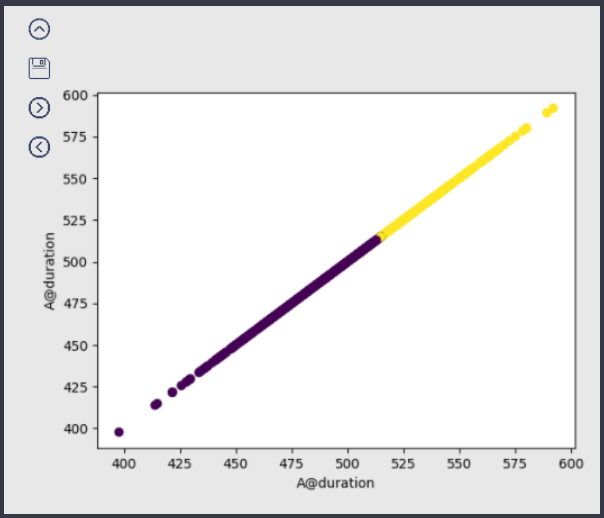

Figure 4: **KNN**

Authors:
Dyllan Cartwright
Radu Sterie
Arash Yadegari
(University of Groningen)
