# Walkthrough GPX

Dieses Dokument soll den Output des unten gezeigten Codes näher beschreiben.

This script demonstrates how to use the gp\_explainer library for explaining the predictions made by machine learning models, such as Decision Tree, Neural Network, and Random Forest. It uses three different datasets (moons, circles, and make\_classification) to fit and visualize the models and their explanations.

Here's a summary of the key parts of the script:

* Import necessary libraries, such as NumPy, Matplotlib, scikit-learn, and gp\_explainer.
* Define classifiers as a dictionary, including Decision Tree, Neural Network, and Random Forest.
* Define datasets as a dictionary, including moons, circles, and make\_classification.
* Create a figure for the subplots using Matplotlib.
* Iterate over each dataset and classifier to fit the model, create the explanation, and plot the results:
  + Preprocess the dataset using StandardScaler and split it into train and test sets.
  + Fit the model using the train set.
  + Create a Gpx object for the model's predict\_proba method, and generate an explanation for a randomly chosen instance from the test set.
  + Plot the dataset, the model's decision boundaries, the gp\_explainer's interpretation, and a visualization of the math expression tree generated by gp\_explainer.
* Save the final figure containing all the subplots as an SVG file.

This script allows you to visualize the predictions made by the machine learning models and the explanations provided by gp\_explainer, as well as compare their performance on different datasets.

Repository:  
A screenshot of a computer

Description automatically generated with medium confidence

A screen shot of a game

Description automatically generated with low confidence

The final output of this code consists of a grid of subplots for each dataset and classifier. Each row of subplots represents a dataset, and for each dataset, there are four columns of subplots for each classifier, showing:

1. The training dataset.
2. The model's decision boundaries.
3. The gp\_explainer's interpretation of the decision boundaries.
4. The gp\_explainer's math expression tree visualization.

For the first row of subplots, the 'moons' dataset is used, and the following steps are taken for each classifier:

1. The dataset is standardized using StandardScaler, and then it's split into training and testing sets with an 80-20 ratio.
2. The model is trained on the x\_train and y\_train data.
3. A Gpx object is created for the model's predict\_proba method using the training data, with num\_samples=100 and k\_neighbor=20.
4. A random instance from the test set is selected, and the gp\_explainer generates an explanation for it.
5. The decision boundaries of the model and the gp\_explainer's interpretation are visualized using contour plots.

Let's break down the first row of subplots:

* First column: The training data for the 'moons' dataset is plotted. The data points are color-coded based on their classes.
* Second column: The decision boundaries of the classifier (e.g., Decision Tree) are plotted. The background contour plot shows the classifier's decision regions, while the colored data points represent the training data. The green 'X' marker represents the instance for which the explanation is generated.
* Third column: The gp\_explainer's interpretation of the decision boundaries is plotted. The background contour plot shows the interpreted decision regions, and the colored data points and green 'X' marker are the same as in the second column.
* Fourth column: The gp\_explainer's math expression tree visualization is shown as an image. This tree represents the mathematical expression that gp\_explainer uses to approximate the model's decision boundaries.

The numbers in the last column (math expression tree) can be interpreted as follows:

* The numbers at the leaf nodes represent the output of the mathematical expression for a specific path through the tree.
* The numbers at the internal nodes represent the coefficients or parameters used in the mathematical operations at that node.
* The structure of the tree and the numbers together form a mathematical expression that approximates the model's decision boundaries.

In summary, the final output of this code is a grid of subplots that visually compares the performance of different classifiers on various datasets. It also shows how gp\_explainer can be used to provide interpretable explanations for complex decision boundaries generated by these classifiers.

