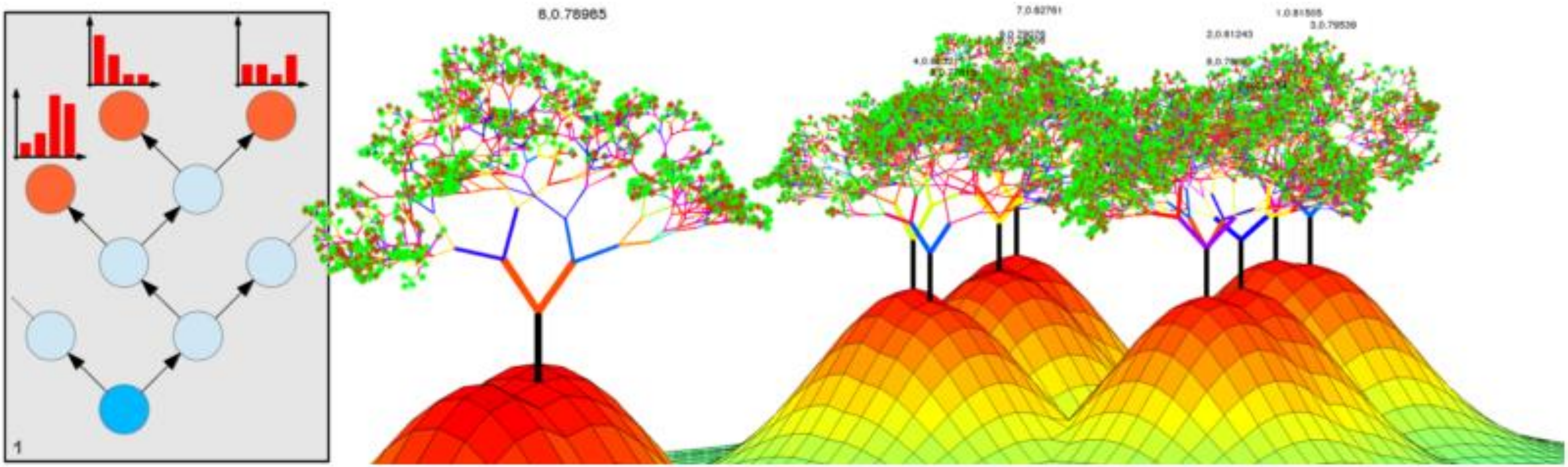
Random Forest from scratch

interpreting the Math behind the 'Black Box'

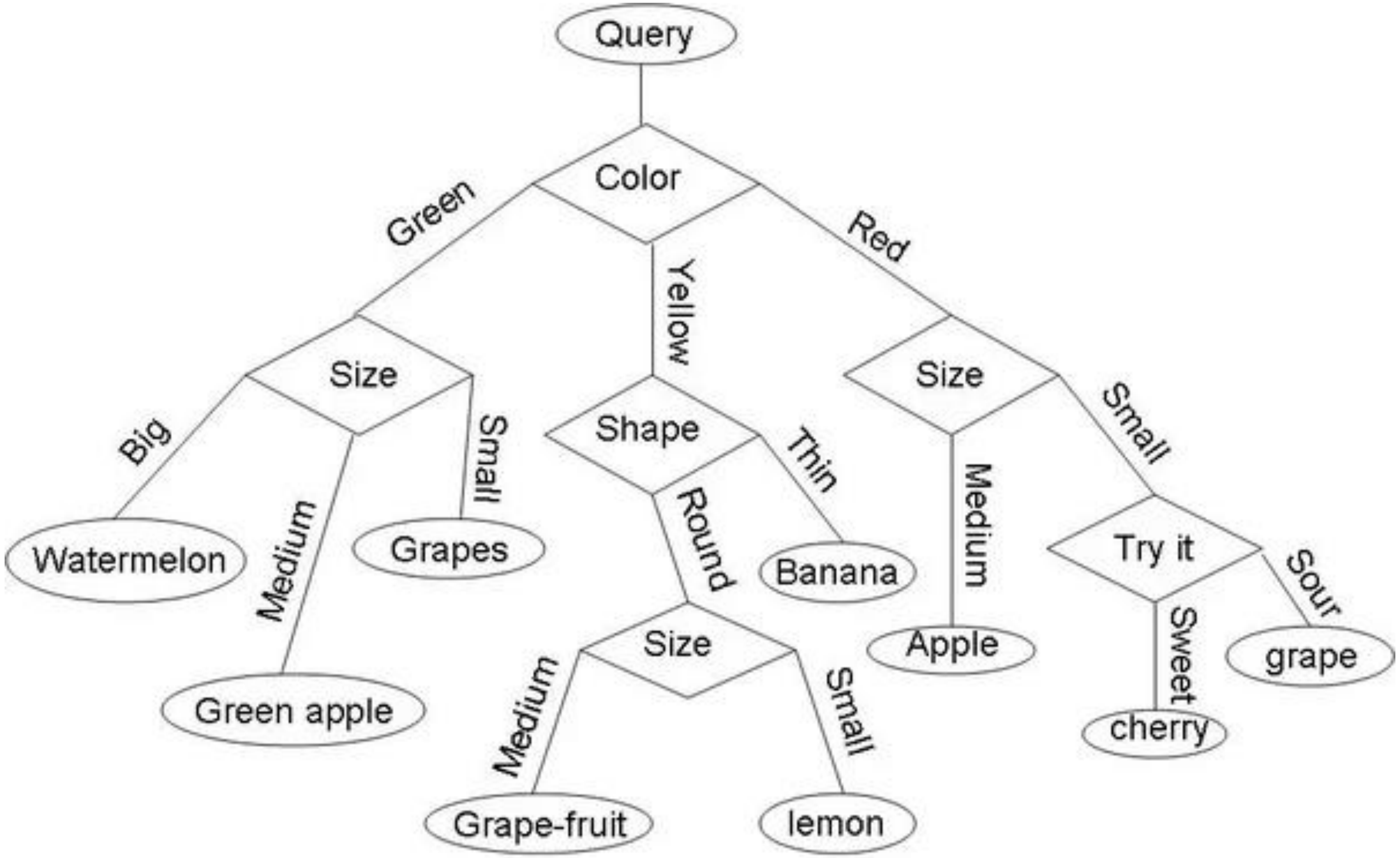
Motivation

Random Forest Ensembles are widely used for real-world machine learning problems, **Classification** as well as Regression. Their popularity can be attributed to the fact that practitioners often get **optimal results** using a Random Forest algorithm with **minimal data** cleaning, and no feature scaling.



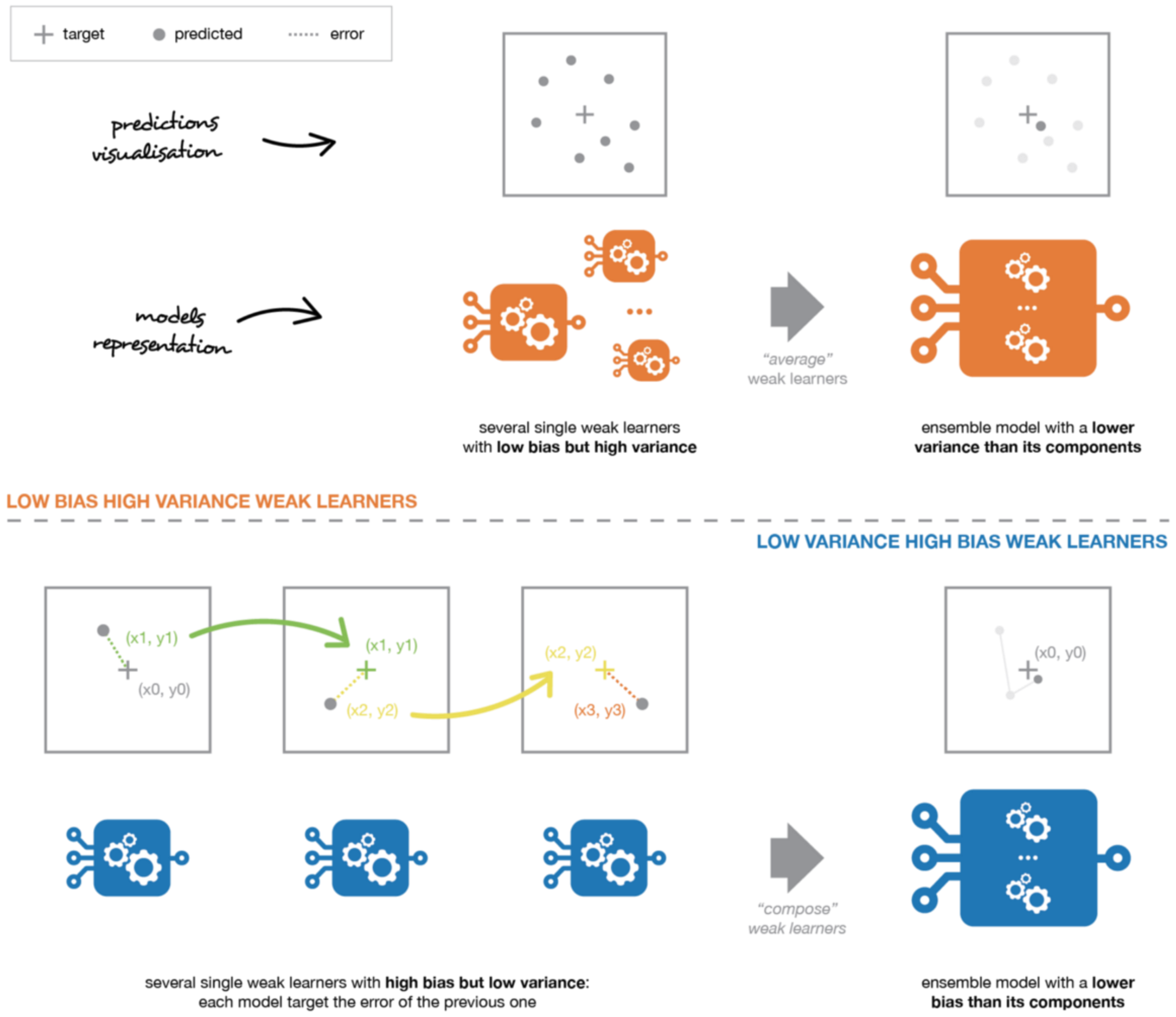
Motivation

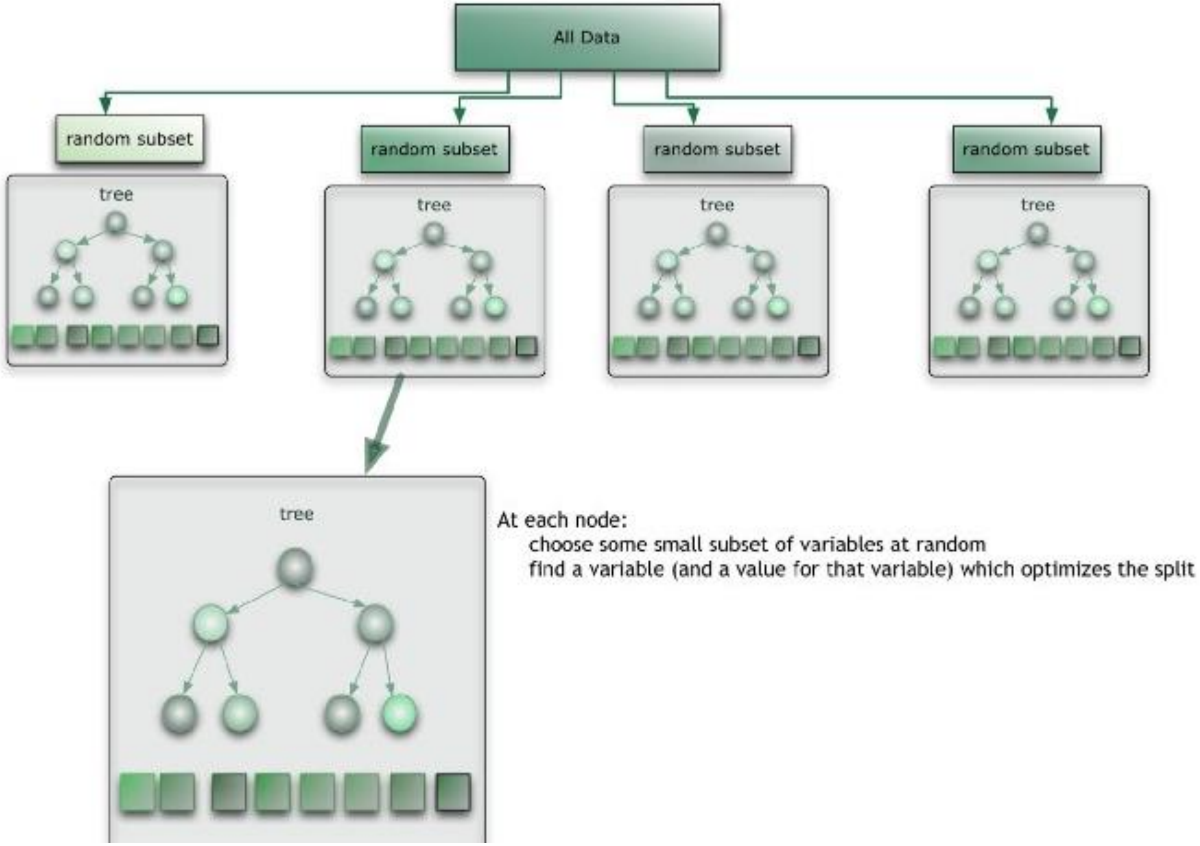
Widely believed to be a black box, a quick walk-through of the algorithm will prove it is actually quite interpretable, apart from being a powerful technique leveraging the **‘power of the majority vote’.**



Introduction

- Random Forest Ensembles are **a *divide-and-conquer approach*** used to improve performance of individually weak Decision Tree models.
- The main principle behind this is that a group of **"weak learners"** can come together to form a "strong learner". Each classifier, individually, is a "weak learner," while all the classifiers taken **together are a "strong learner".**

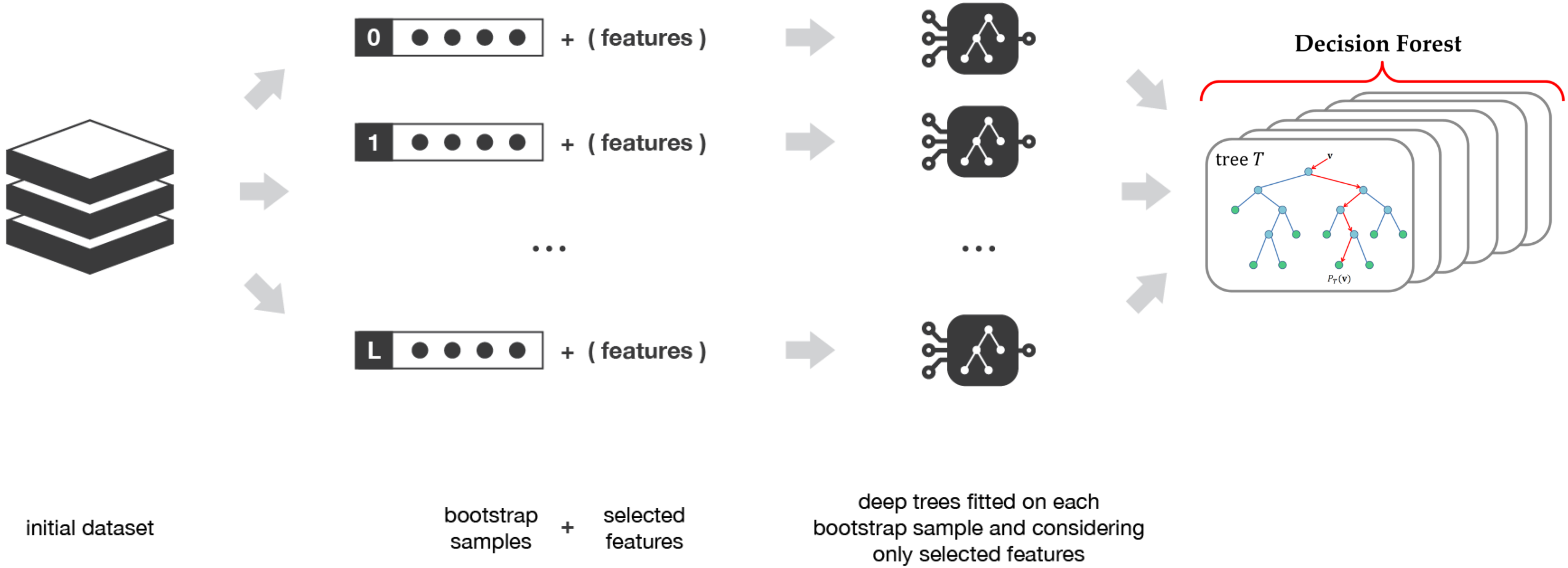


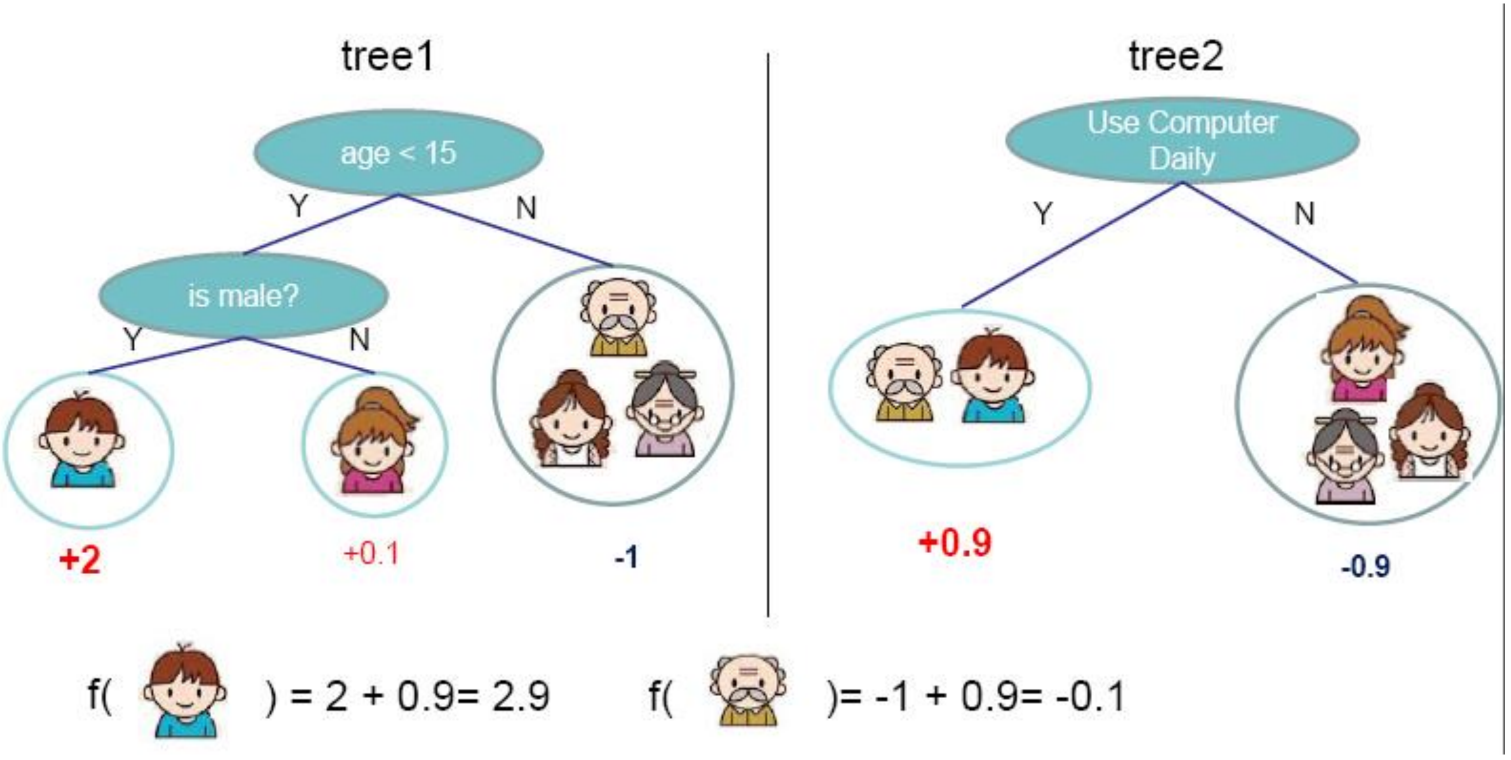


Random Forest Ensemble

- At the heart of the Random Forest concept is averaging the results from a number of Decision Trees.
- Decision Trees are often handy tools to explain the intuition behind a prediction to people unfamiliar with Machine Learning. But explaining how a Random Forest arrived at a prediction, and using which features or independent variables, can be quite a task. Random Forests are often misinterpreted as ‘Black Boxes’ or difficult to understand.

Decision forest



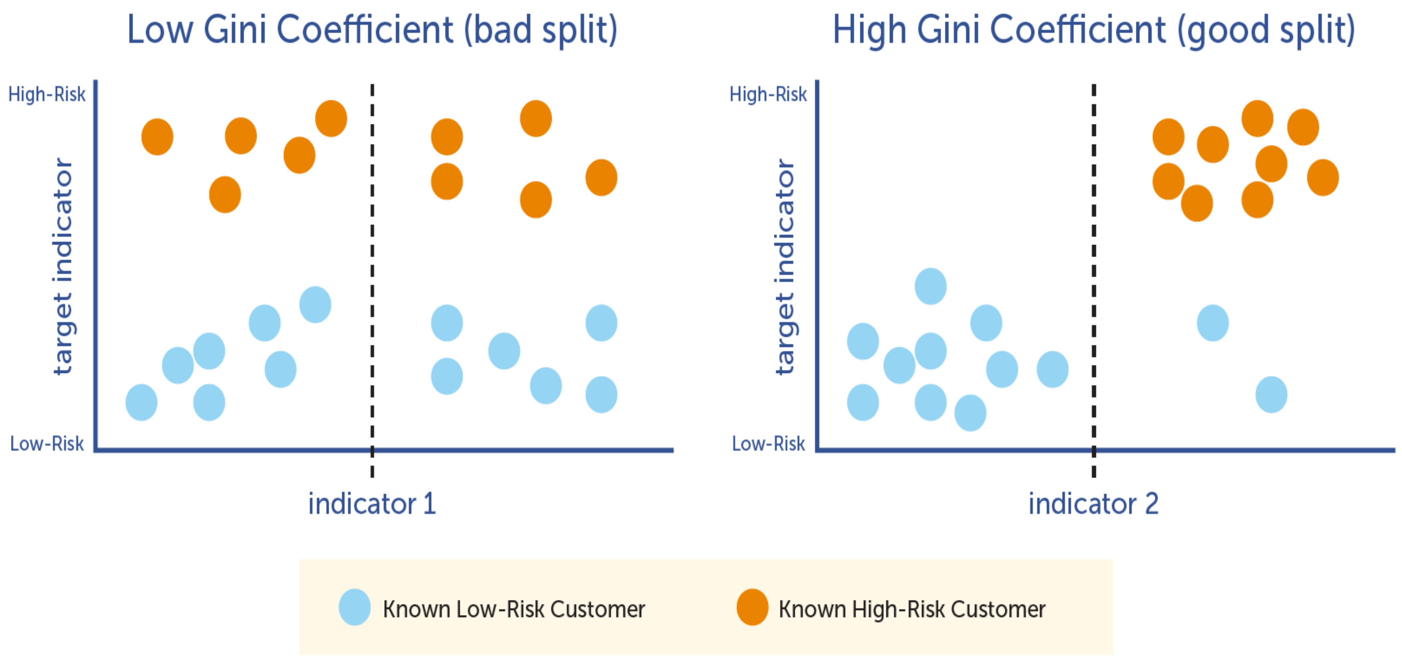


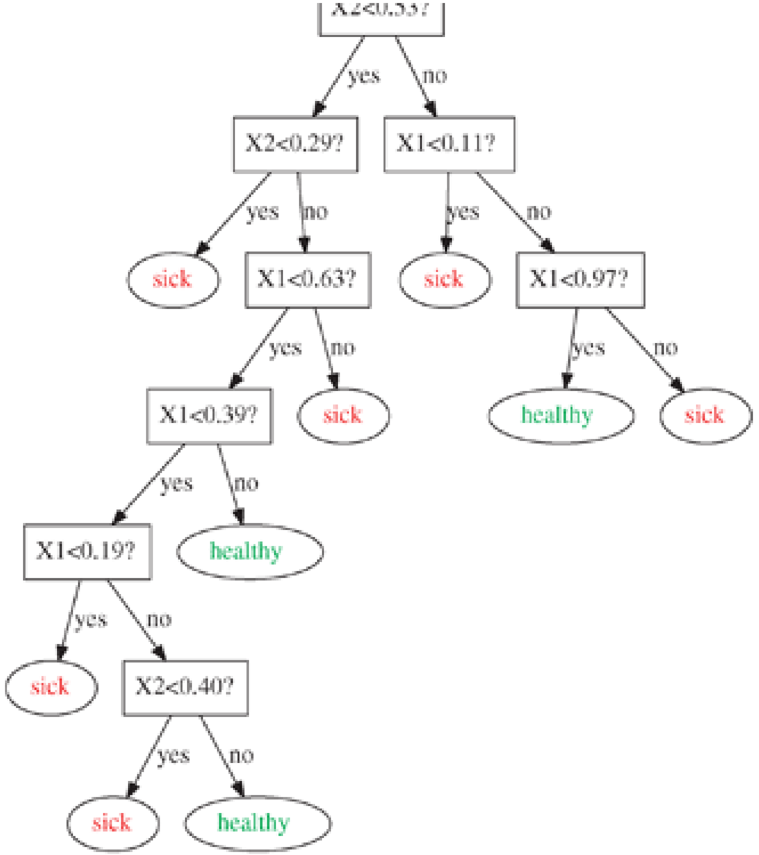
What is a Decision Tree?

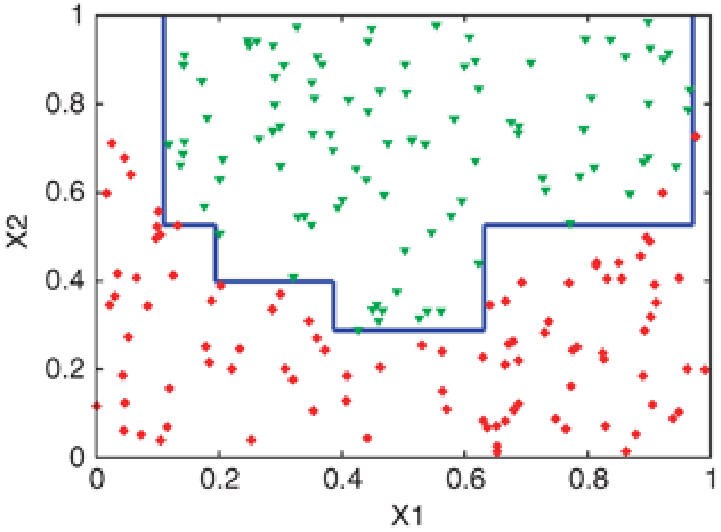
- Decision tree is a simple, deterministic data structure for modelling decision rules for a specific classification problem.
- At each node, one feature is selected to make separating decision. We can stop splitting once the leaf node has optimally less data points.
- Such leaf node then gives us insight into the final result (Probabilities for different classes in case of classfication).

How does it split?

- The most decisive factor for the efficiency of a decision tree is the efficiency of its splitting process. We split at each node in such a way that the resulting **purity** is maximum.
- Well, purity just refers to how well we can segregate the classes and increase our knowledge by the split performed. An image is worth a thousand words. Have a look at the image below for some intuition:







Visualization

- Each split leads to a straight line classifying the dataset into two parts. Thus, the final decision boundary will consist of straight lines (boxes).
- Each split leads to a straight line classifying the dataset into two parts. Thus, the final decision boundary will consist of straight lines (or boxes).

Easy use of Random Forest Classification

dzetsaka : Classification tool

Fast and Easy Classification plugin for Qgis

Plugin for semi-automatic classification with Gaussian Mixture Model, Random Forest*, and SVM* classifiers.
Very easy and fast to use.
*You need to install scitkit-learn library to use these algorithms.
For more information on this tool check our github :
https://github.com/lennepkade/dzetsaka/

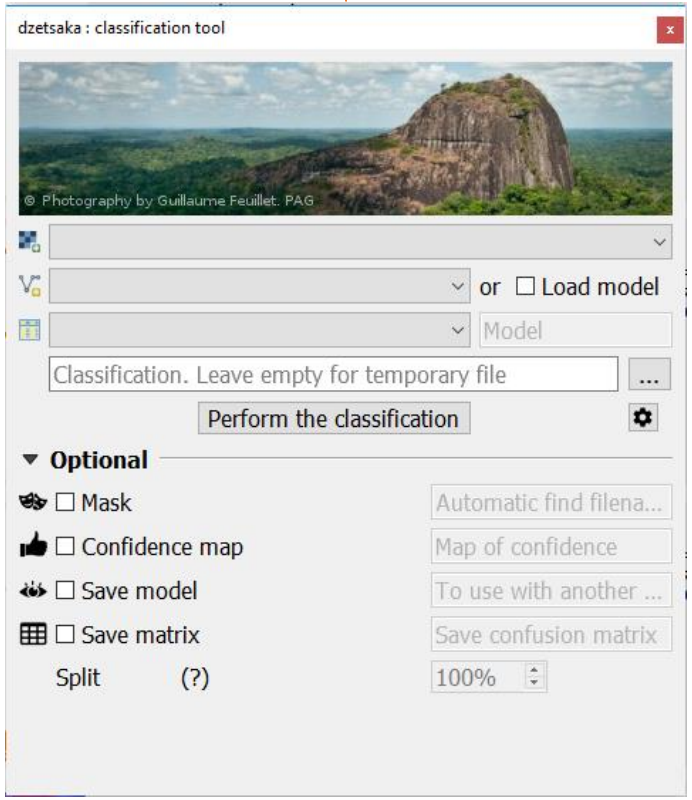
35 rating vote(s), 43805 downloads

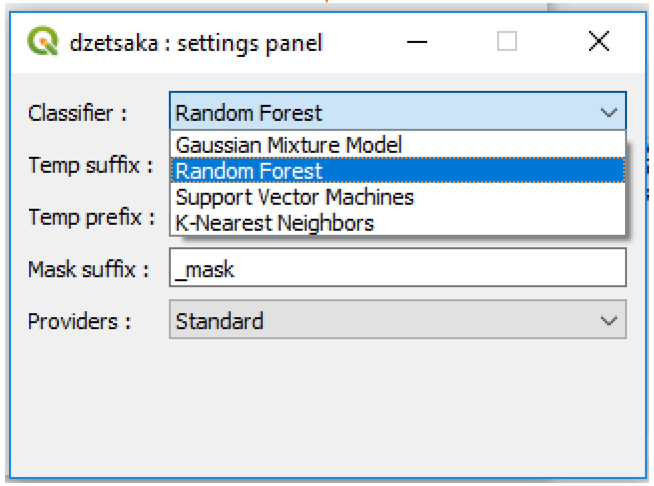
Category	Raster
Tags	classification,semi-automatic,gaussian,mixture,model,random forest,svm,knn,forest,processing
More info	homepage bug tracker code repository
Author	Nicolas Karasiak
Installed version	3.4.8
Available version	3.4.8
Changelog	3.4.8 * Fix errors when number of classes > 44 (problem of datatype in sample extraction). 3.4.7 * Support more than 255 classes to predict (if n > 255, raster datatype will be set to uint16) 3.4.6 * Minor fixes and remove SLOO training due to error in code 3.4.5 * Fix bug when predicting a raster with a previous model and no vector loaded in Qgis. 3.4.4 * Remove install of sklearn with python pip (causes bugs).

Initially based on Gaussian Mixture Model classifier developped by Mathieu Fauvel (now supports Random Forest, KNN and SVM), this plugin is a more generalist tool than Historical Map which was dedicated to classify forests from old maps. This plugin has by developped by Nicolas Karasiak.

Dzetsaka

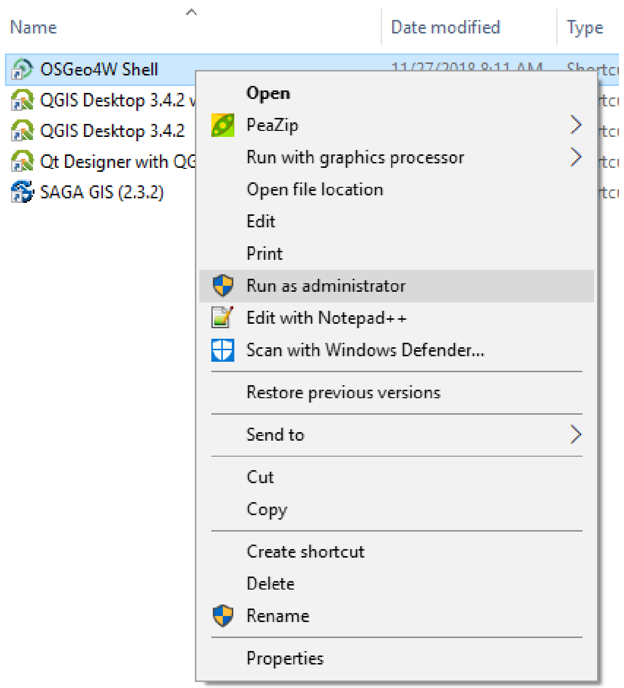
- After plugin instalation





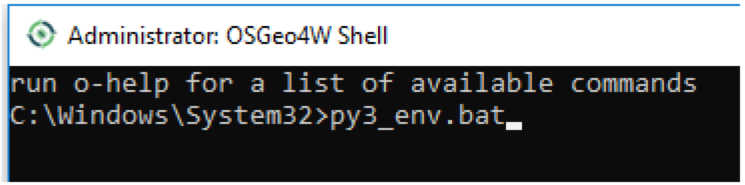
Dzetsaka

- To install Random Forests

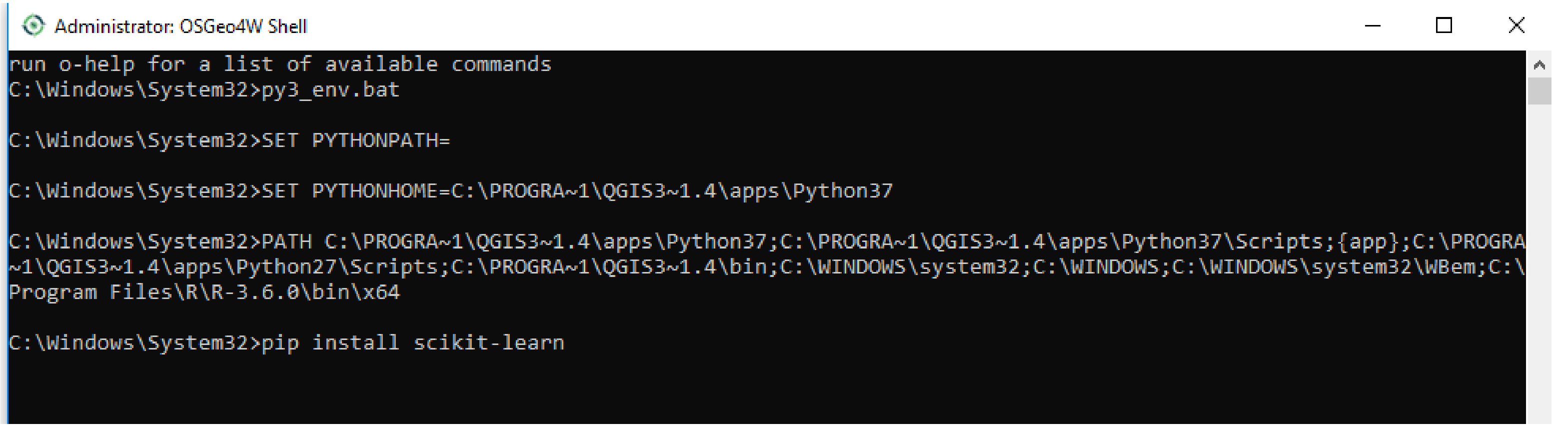


1. Open OsGeo shell in admin

2. Run command: py3_env.bat



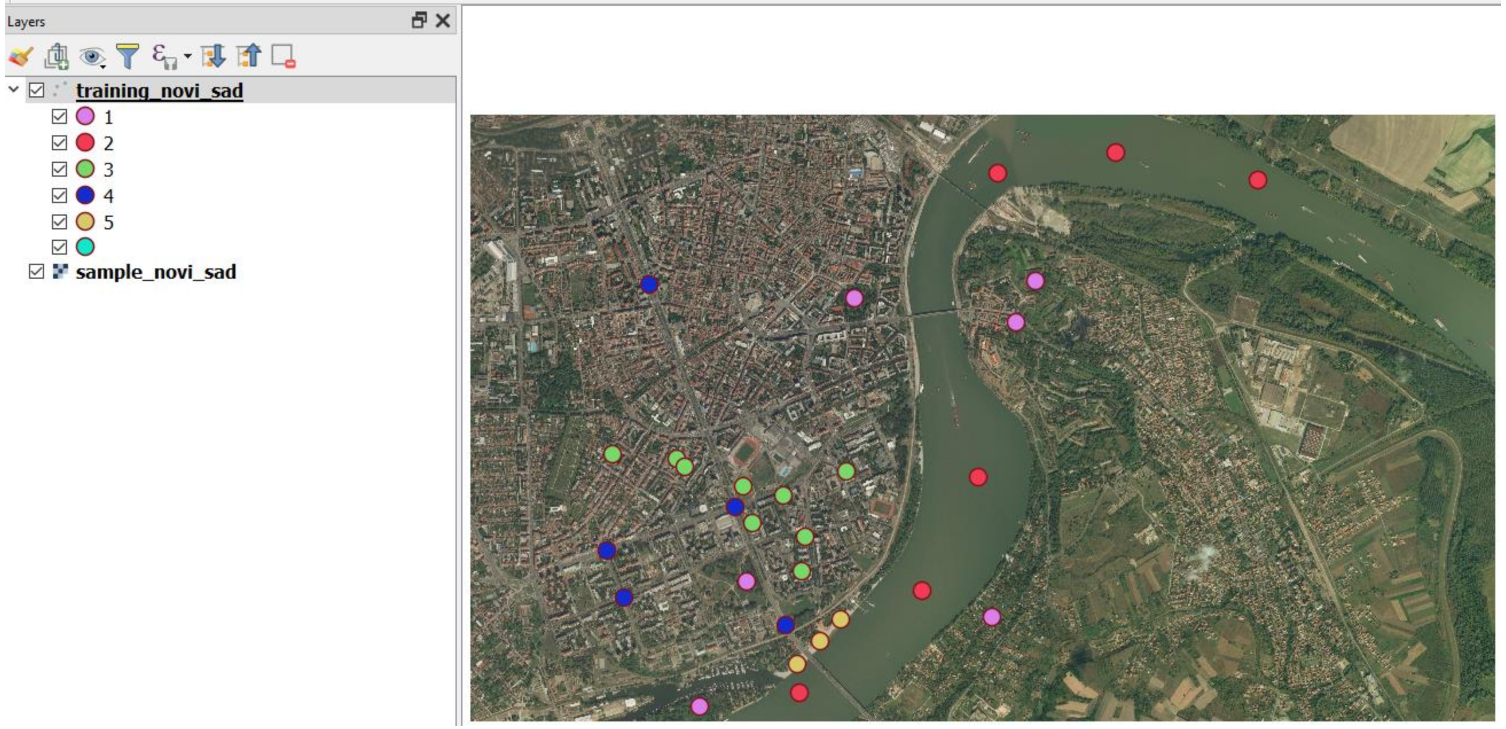
3. Run command: pip install scikit-learn



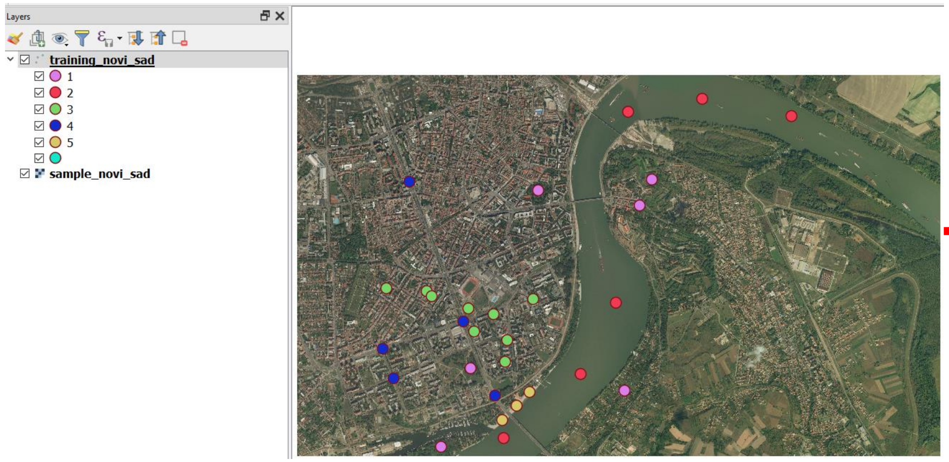
How to run Dzetsaka tool

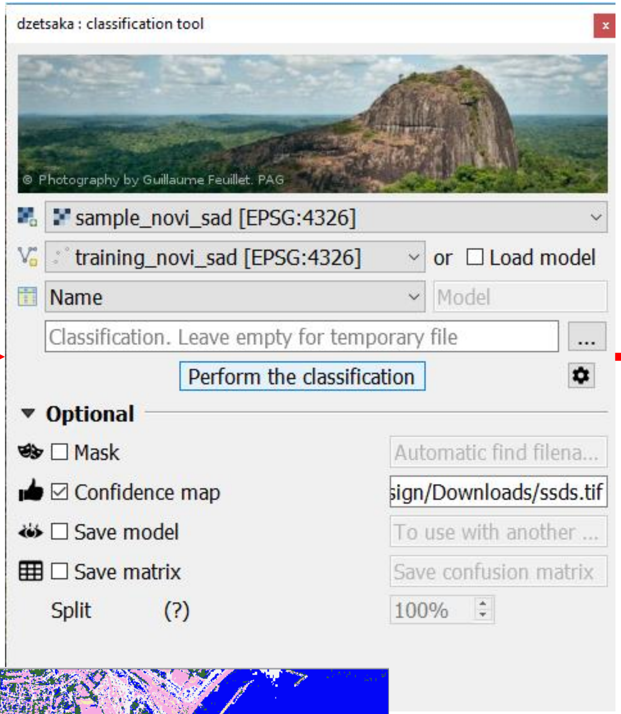
- You need 1 raster and 1 shapefile
- So you need to create a **shapefile with a numeric column** where you save your classification number for each polygon. Here's a example

Class	Type
1	Forest
2	Rock
3	Water
4	clouds
5	Shadow

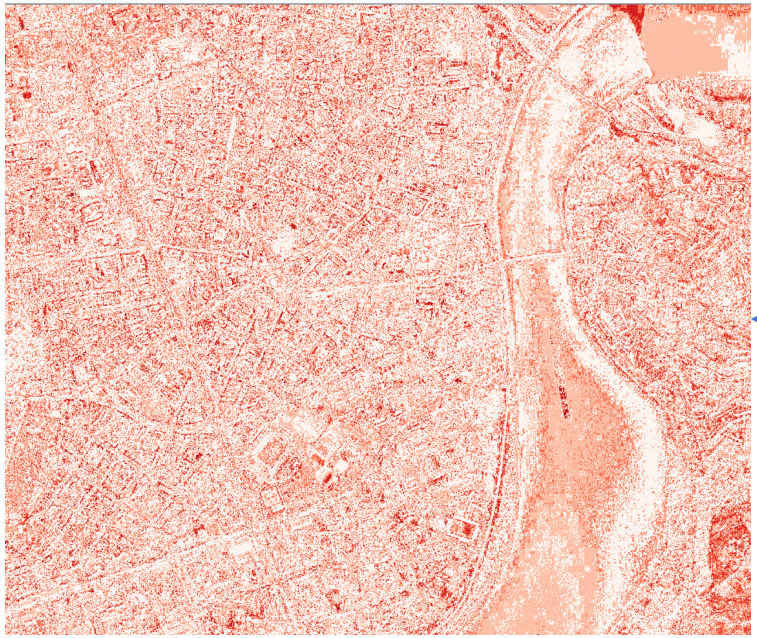


How to run Dzetsaka tool





Confidence map



Classification map