

# Martin's Blog

Research on Human-Computer Interaction and  
Ubiquitous Computing

## TAG: RANDOMFOREST

**JUNE 30, 2015**

### Exporting RandomForest Models to Java Source Code

This post shares a tiny toolkit to export WEKA-generated Random Forest models into light-weight, self-contained Java source code for, e.g., Android.

It came out of my need to include Random Forest models into Android apps.

Previously, I used to use Weka for Android. However, I did not find a way to export a Random Forest model in a way that my apps can load it reliably across devices, so the apps had to compute the model on each start — which can take minutes.

`androidrf` solves the problem in a simple way: a python script parses the console output of WEKA when training a RandomForest model with the `-printTree` option enabled. Then, it creates a single Java source file implementing those trees with simple if-then statements.

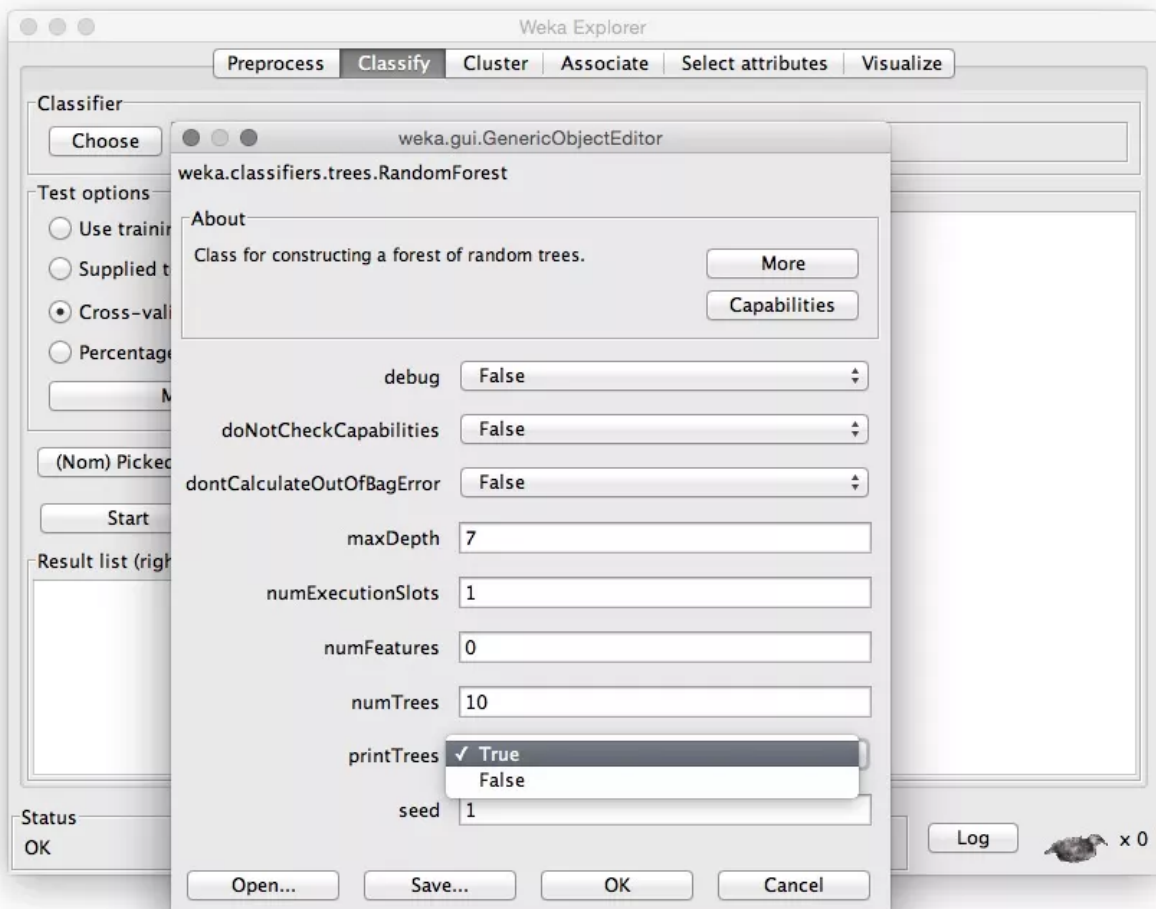
The library ships with three additional Java classes that allow to run and test the generated classifiers.

The code is available on Github under the MIT Licence: `androidrf`

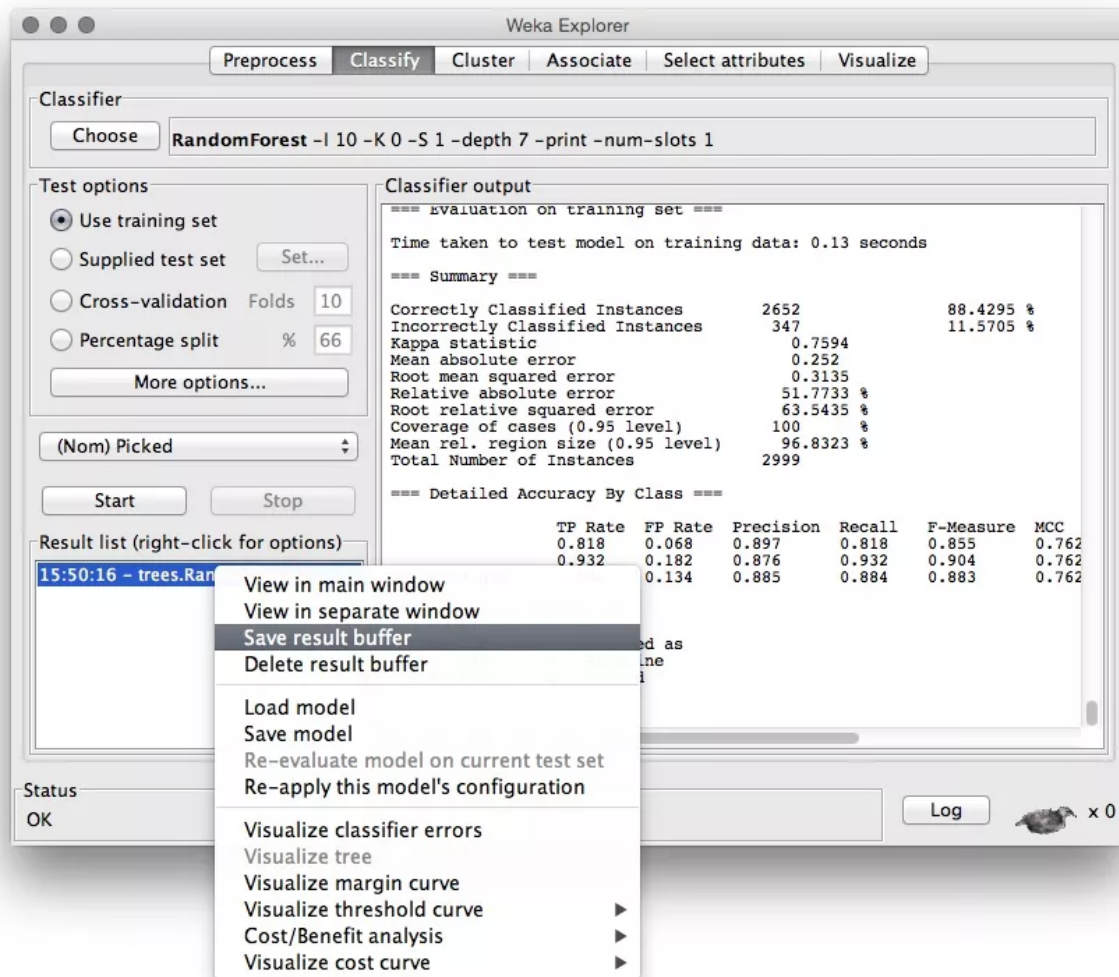
### How to use it

(for people who are familiar with WEKA):

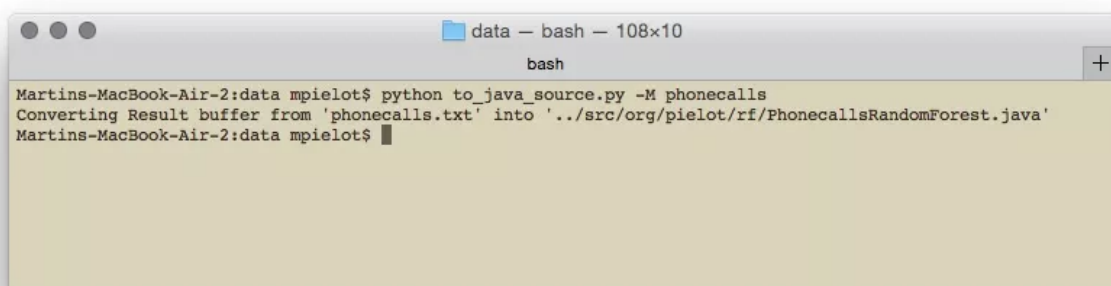
Load your data set into WEKA, choose RandomForest as classifier, and enabled the 'printTrees' option for your RandomForest classifier. Hint: limit the depth of the trees with the 'maxDepth' option, because otherwise the resulting source files may become huge.



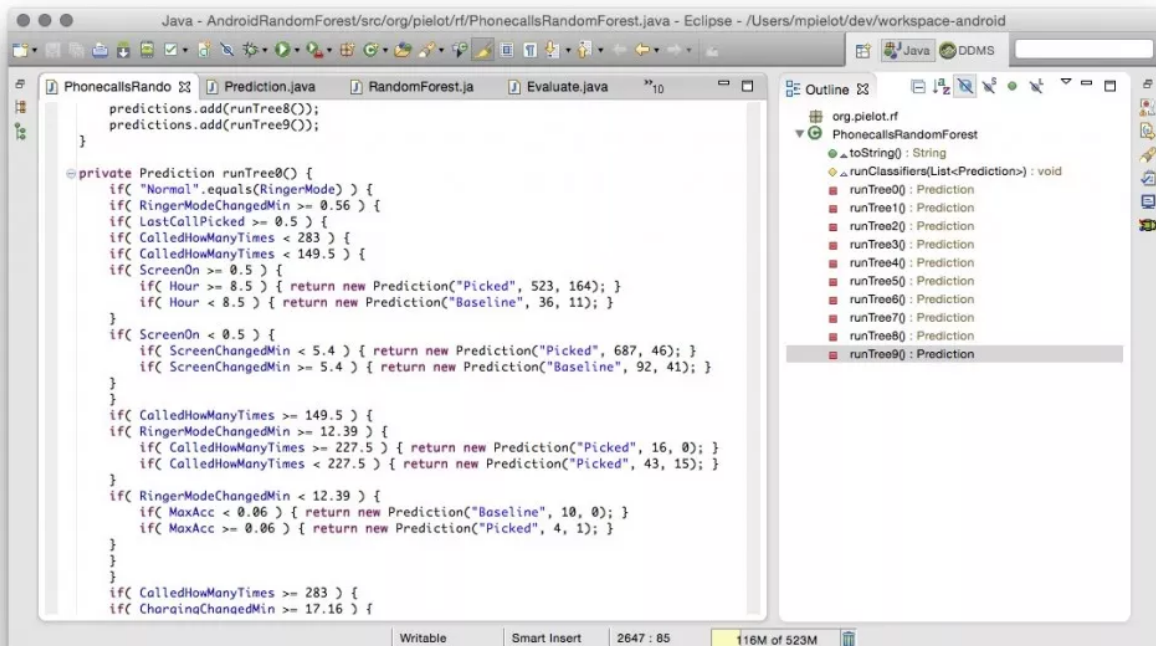
Save the output of the results buffer into a .txt file. Best, save it into the 'data' folder of the *androidrf* project.



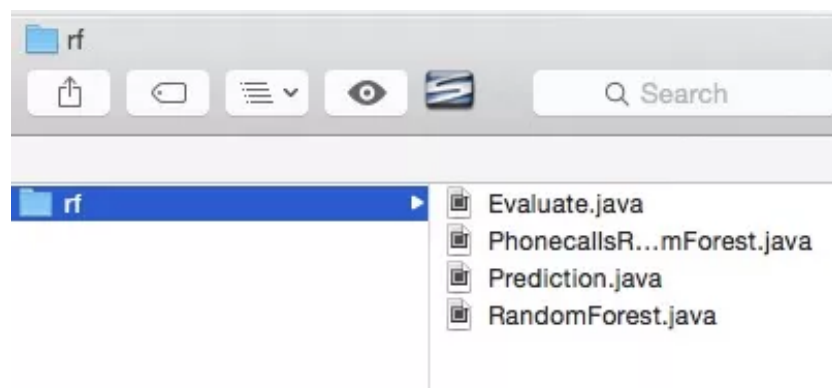
Open a terminal, enter the 'data' folder of the androidrf project, and execute `python to_java_source.py -M filename (without.txt)`.



A class with the name `FilenameRandomForest` should appear in `androidrf/src/org/pielot/rf`



All you need to do is to copy the Java class together with the three pre-existing Java classes (Prediction, Evaluation, RandomForest) into your project. It should compile without error.



The features have been added as fields to your classifier. Hence, in order to specify the features, simply populate those fields. Then, run `runClassifiers(List predictions)` to obtain a `Prediction` with the details of the prediction (predicted class, certainty, ..).

Voila! You have a light-weight, portable, working Random Forest model.

Share this:



