# Read me file for running Weka Random Tree and Python SVD

## Weka Random Tree

“/mounts/idir-server5/proj/rohitb/classifiers/custom\_random\_forest/src” this location has the setup required to run the code

“/mounts/idir-server5/proj/rohitb/classifiers/custom\_random\_forest/src/random\_forest” directory has two files

* “RandomForest.java” can be used to build the classifier and save them in the file
  + Requires ‘data’ directory to be created at the same level as directory that contains these files

Eg: “/mounts/idir-server5/proj/rohitb/classifiers/custom\_random\_forest/src” contains “data” directory and “trees” directory should be a subdirectory of the “data”

* + Full path of CSV file containing the data needs to be changed in “RandomForest.java”

Eg: “String file = "/mounts/idir-server5/proj/rohitb/data\_112.csv";”

* + cd to “/mounts/idir-server5/proj/rohitb/classifiers/custom\_random\_forest/src” and Compile.sh & run.sh to compile and run the code
  + “buildClassifiers(100)” method should be called to generate the trees, number of trees to be generated should be the input parameter to the function.

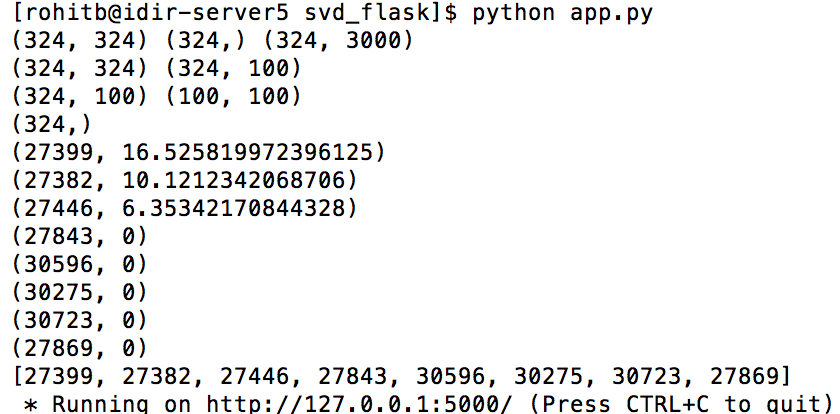
It will read data from the input file and build the number of trees passed in the “buildClassfiers” call and save those trees in “data/trees” and other required data in “data”

* “RandomForestPredict.java” file can be used to read those saved classifier and perform classification
* “loadData();” call will load all the required data and “classifyData(<>, <>)” to classify and rank the candidates. “classifyData(<>, <>)” takes two parameters
  + First Array List parameter is the user history
  + Second Array List parameter is the candidates for ranking

There is an example in “RandomForestPredict.java” main function, which calls loadData and ClassifyData functions

## Python SVD

* Installing requirements
  + easy\_install pip
  + pip install numpy
  + pip install scipy
  + pip install sparsesvd
* Cd to “/mounts/idir-server5/proj/rohitb/classifiers/svd\_flask” and “train.py” and “app.py” contains all the required code.
* Create a “data” directory next to app.py
* “train.py” can be used to build and save the model.
* “app.py” can be used to run the webserver which reads the saved model.
* Change the csv file name in “app.py” and run “python app.py” to start the server. It will train and the webserver starts.



The server will be accessible only from the local machine and it runs on port 5000

* Code for calling this web service from java code is in “/mounts/idir-server5/proj/rohitb/classifiers/svd\_flask/HTTPRequest” directory.

“apache-commons-httpcore.jar” is required which is in “lib” sub directory and “/mounts/idir-server5/proj/rohitb/classifiers/svd\_flask/HTTPRequest/src/httprequest” has the java file “HttpRequest.java”

Please look at the main function to see how to make a call to python web service.

I haven’t run java code in server5, I just have tried it in my local machine. I have handled of sending the data to the python server and printing the ranked candidates on the console.