**Integrating New Algorithms into Wolf:**

Algorithm takes command line input for parameters associated to algorithm. Also takes as input a path to the file consisting details of previous transaction output and its parameters, which is then used as input for current transaction. Below is the example of an algorithm (Random Forest) to show how an algorithm in WOLF accepts command line parameters.

python RandomForest.py –I output/Splitingdata1/PCA\_result1/SVM\_RFE\_result/splitdatafiles.yaml -o output/Splitingdata1/PCA\_result1/SVM\_RFE\_result/RandomForest\_result1 -d 5 -t 50

-d and –t are algorithms parameters. e.g in this case they represent depth and number of trees. If parameters are not passed by command line, they should be assigned default value by algorithm. It is important to note that algorithm should be able to accept all important parameters.

-i represent input file generated by previous transaction of Wolf. The content of the file are shown in Figure 1. The only thing important in this file is datasets path given under **‘training’** and **‘testing’**. Algorithm should use training data sets to train ML models and then perform predictions on testing data set.

-o give the output folder path of this algorithm. After training, to report results of the algorithm, predictions should be performed for each test data set and the results should be written in separate \*.csv files for each test dataset. In which, 1st column represent original class label for each sample and 2nd column contain corresponding predicted labels for each samples. Finally path of generated \*.csv files should be written in yaml file. So next stage of WOLF should be able to use these \*.csv files.

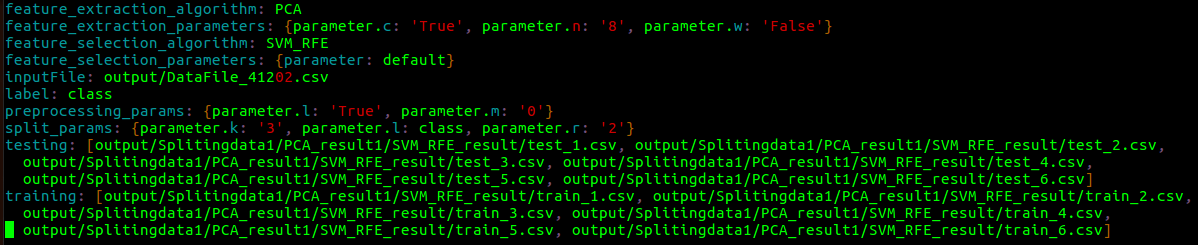


Figure : splitdatafiles.yaml