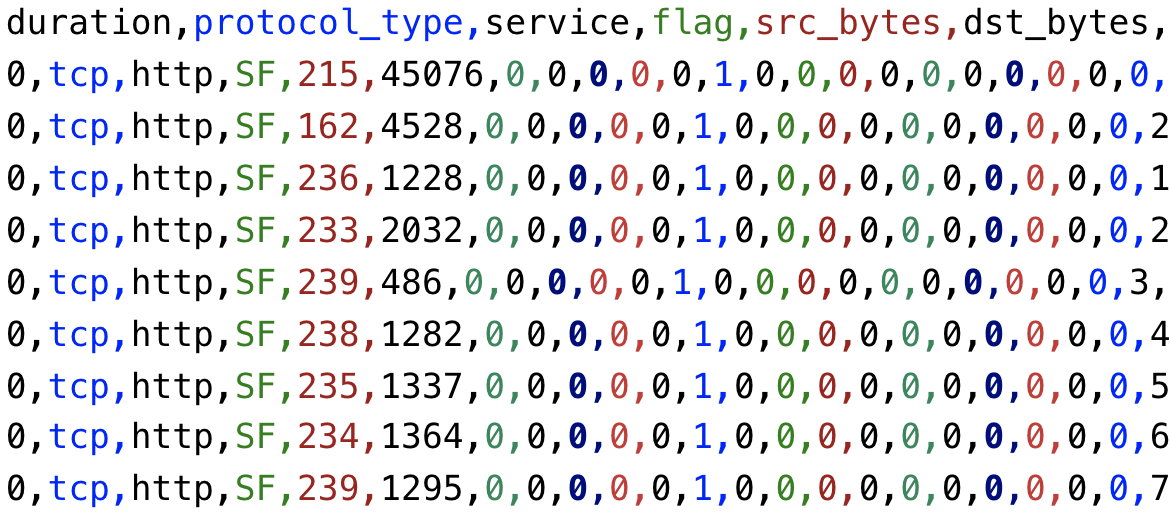
Machine learning cybersecurity **intrusion detection**

# LAB 1: Writing a classifier for kdd99 DATASET

**Lab Description:** This lab is to implement a binary classifier to distinguish normal connections from attacks. You are required to read the data from training set (2,799 records) and test set (1199 records).



You are required to implement it in three ways:

* Using the machine learning software WEKA.
* Writing a python script with the use of the package sklearn
* Writing a python script with the use of the package tensorflow and deep learning techniques.

**Lab Environment:**

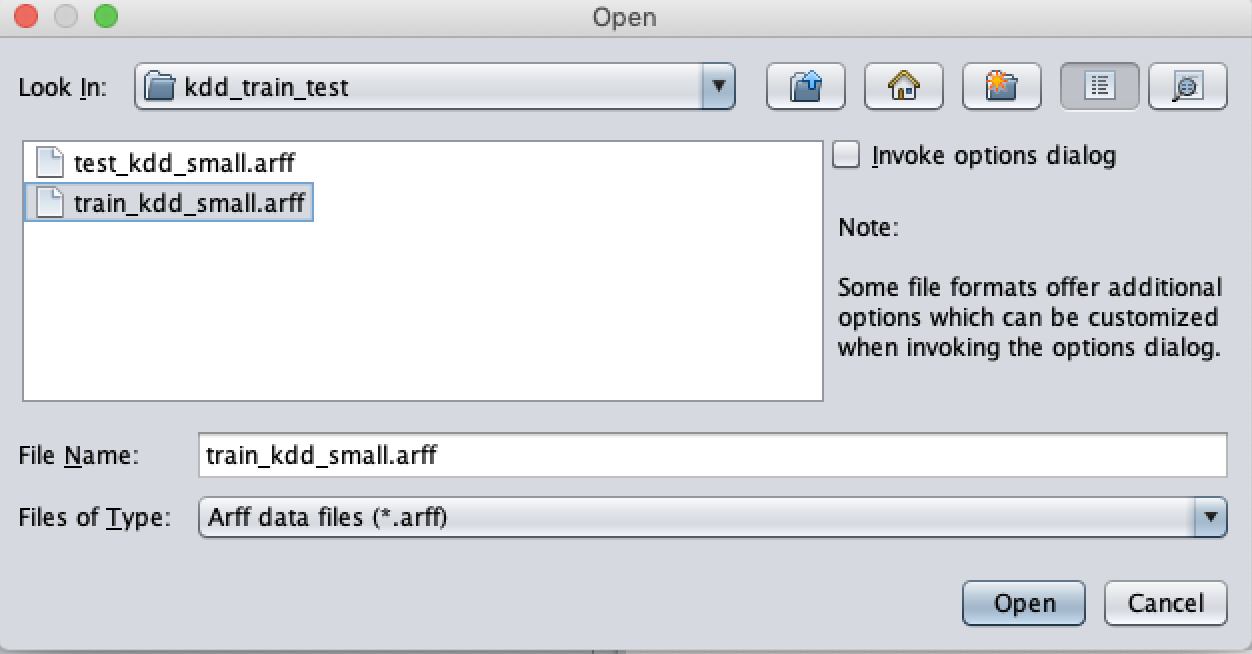
* The students should have access to a machine with Linux system
* WEKA should be installed
* The environment for python is required as well as some packages such as numpy, tensorflow and sklearn.

**Lab Files that are Needed:**

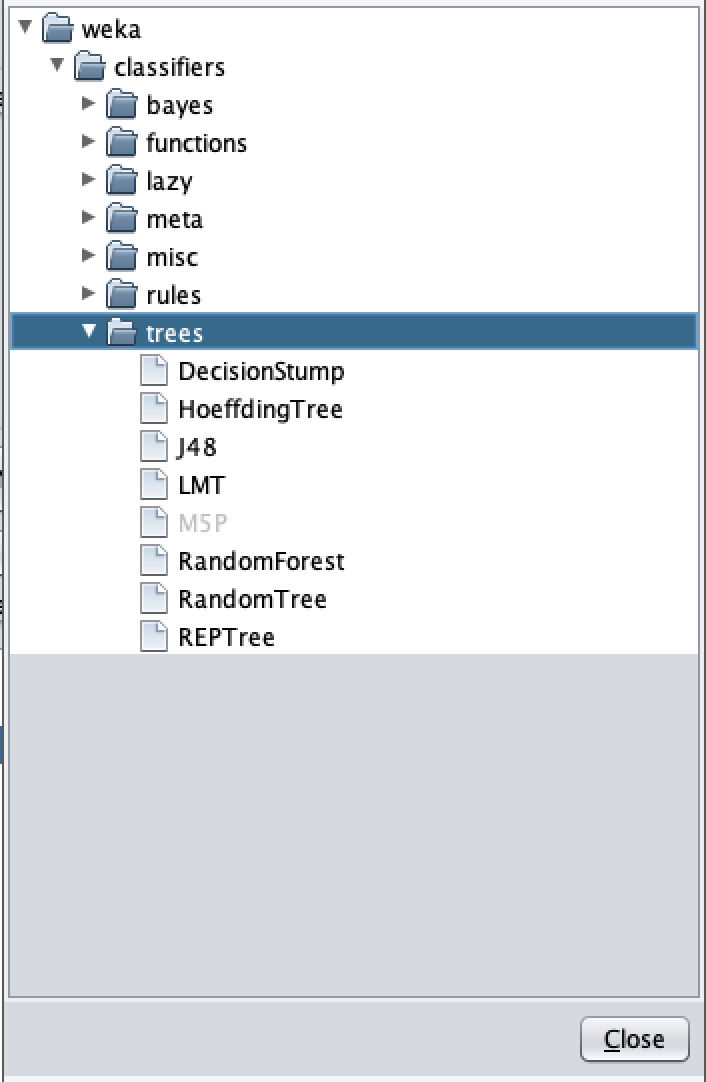
* For this lab you will need four files
* train\_kdd\_small.arff and test\_kdd\_small.arff for WEKA.
* train\_kdd\_small.csv and test\_kdd\_small.csv for python script.
* The last column is the class value, other 41 columns are the features.

### **Lab exercise 1**

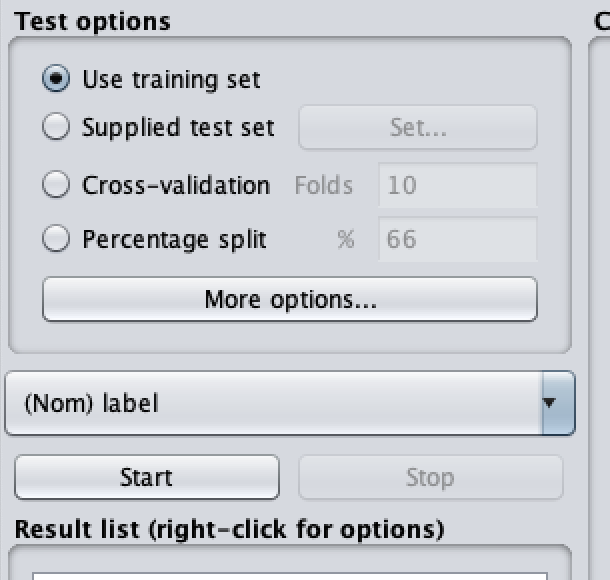
* Import data in the train\_kdd\_small.arff into WEKA (explorer), the files of type should be specified (arff).



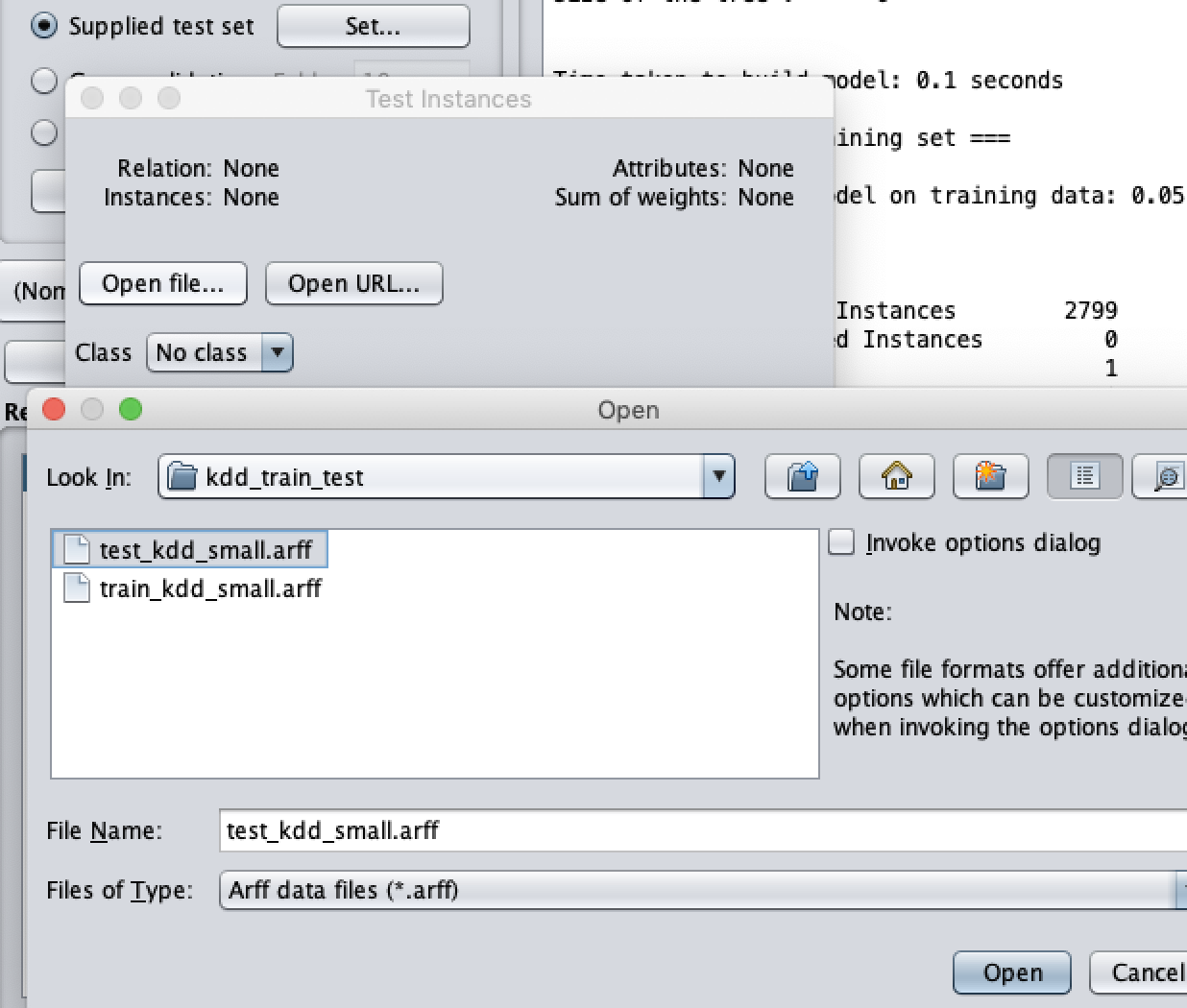
* Choose a proper classifier, such as J48 (Decision Tree)



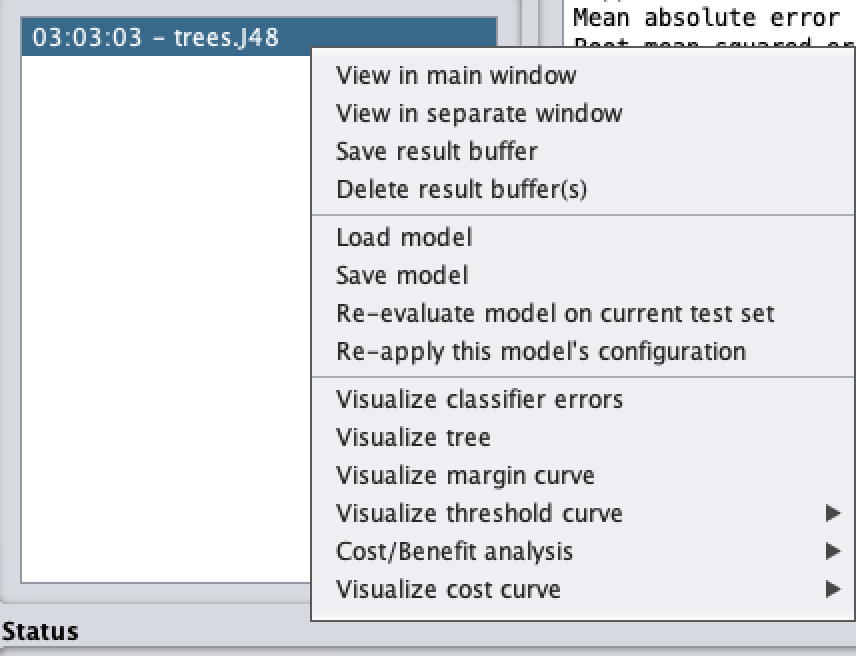
* Specify the test option and the column of class
  + Since we read the data from training set, the test option should be Use training set.



* After the training, supply the test set to evaluate the classifier.

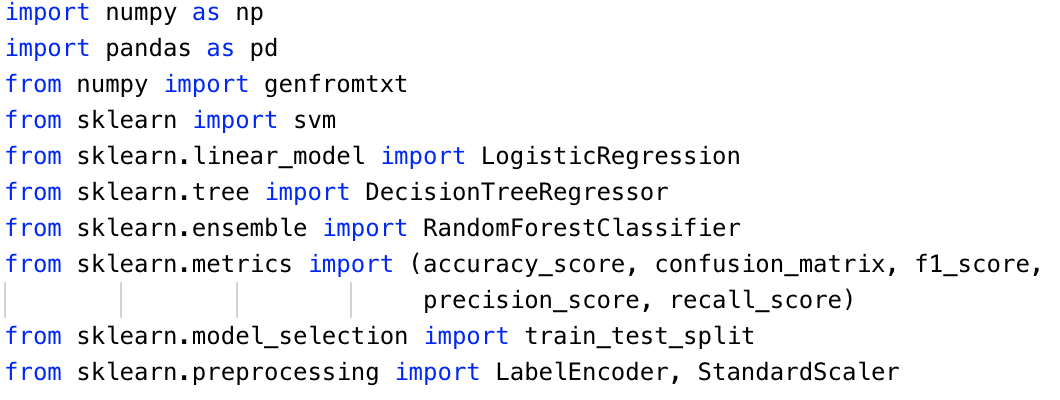


* Right click the model you have trained. Select Re-evaluate model on current test set to perform the evaluation.

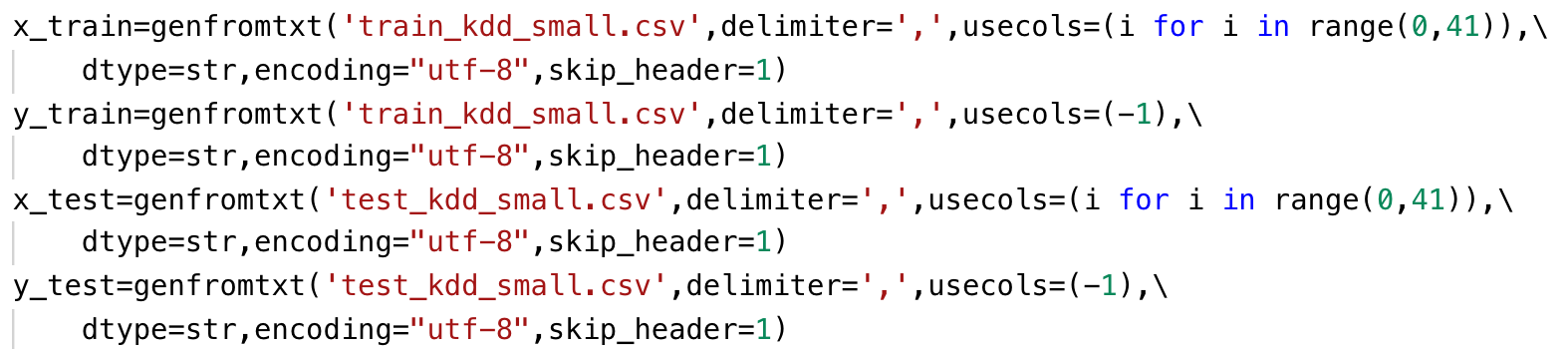


### **Lab exercise 2**

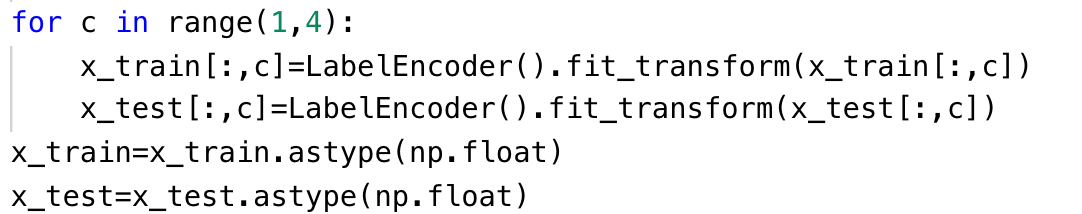
* In this exercise, you need to implement several classifiers with the use of sklearn.
* Import the required libraries



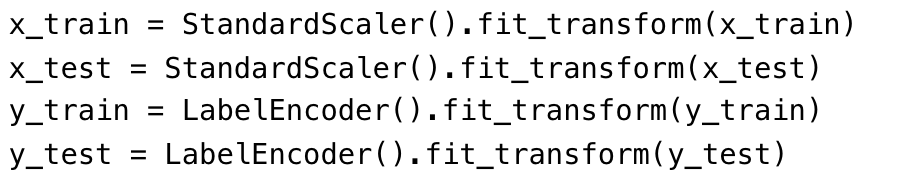
* Read the features and class values from training set and test set with proper method

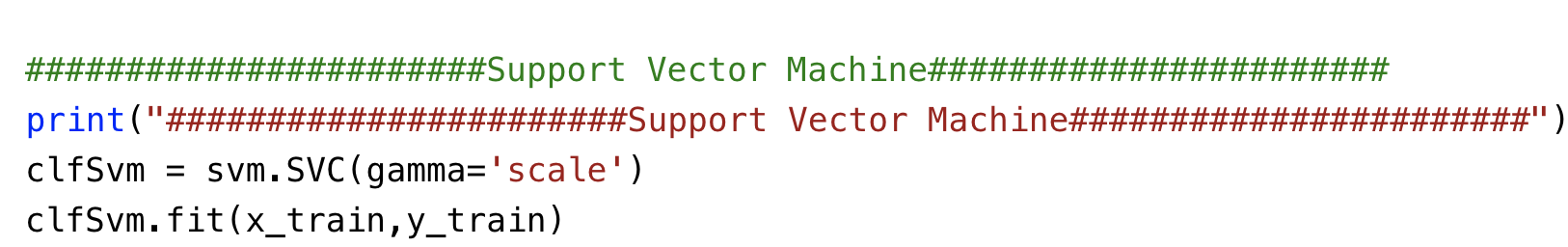


* define x\_train to store the features in training set and y\_train to store class values in training set.
* define x\_test to store the features in training set and y\_test to store class values in test set.
* delimiter indicates the character to split the data in a row.
* usecols indicates which columns will be read. For features, the first 41 columns of the rows will be read. For class values, the last columns of the rows will be read.
* dtype indicates the type of data to read
* Since the first line of the file is names for each column, we set skip\_header to 1 to avoid read the first row.
* Since the second to the fourth features are string type, they are transferred to float type

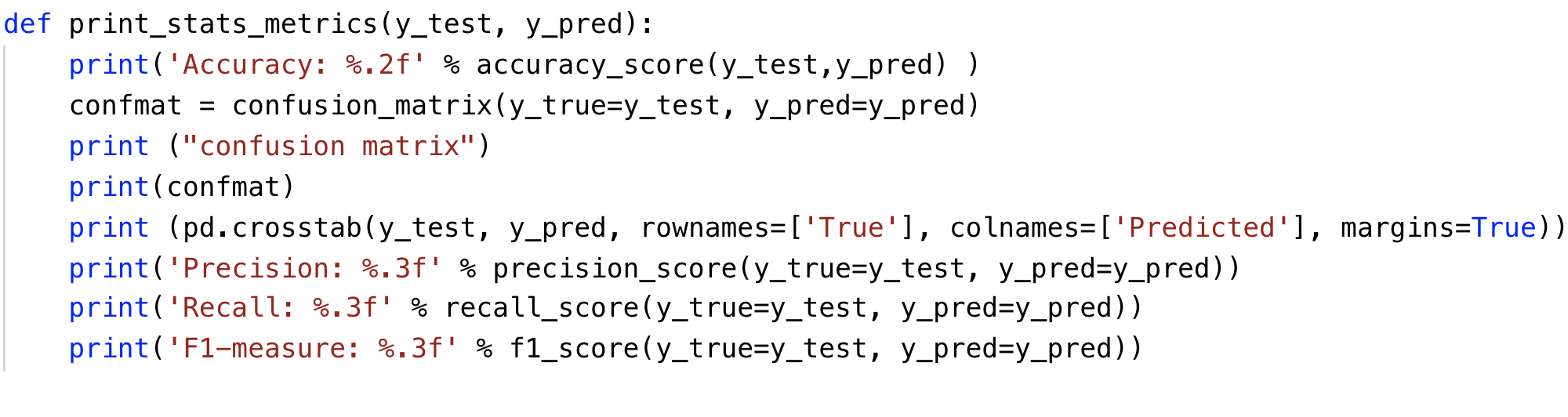


* You may need to create labels for each of kdd classes. When you finish the preprocess step, you can write the python script with the use of sklearn package to build your architecture of classifier.





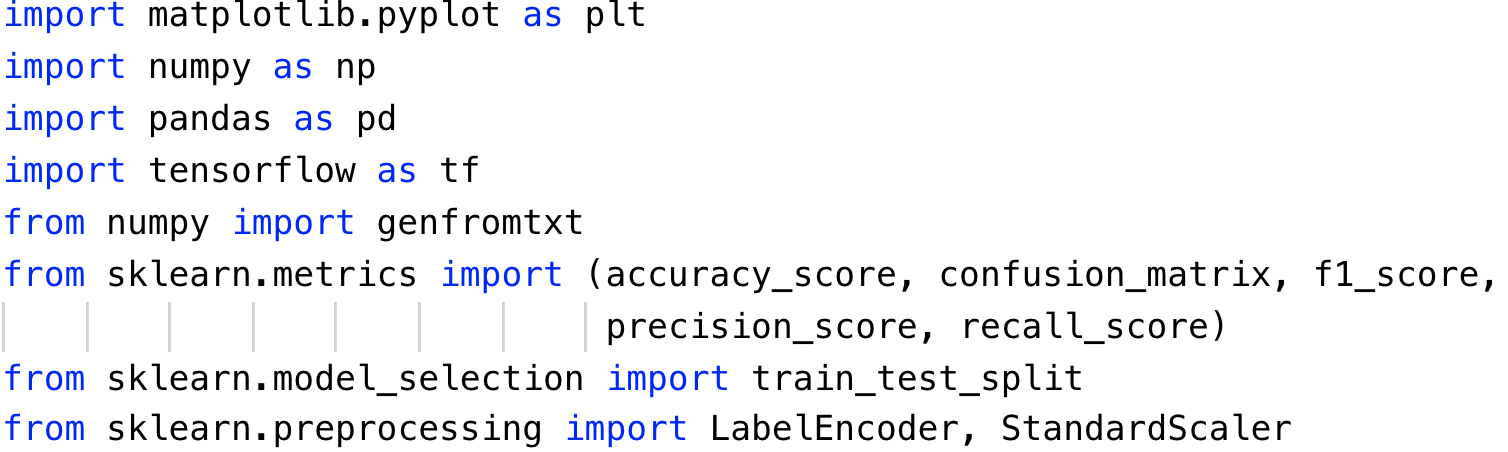
* Please print the statistics metrics such as accuracy, recall, precision and f1 score.



* Implement the classifiers based on Logistic Regression, Support Vector Machine and Random Forest

### **Lab exercise 3**

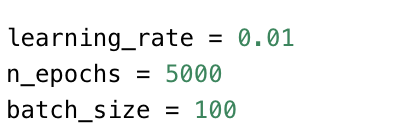
* Use the same data you use in the exercise 2.
* In this exercise, you will implement an artificial neural network classifier based on Tensorflow
* Import the required libraries



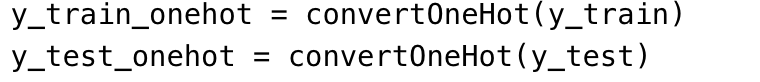
* Repeat the same steps to preprocess the data as Exercise 2. Read the data, standard scale the feature and encode the labels.

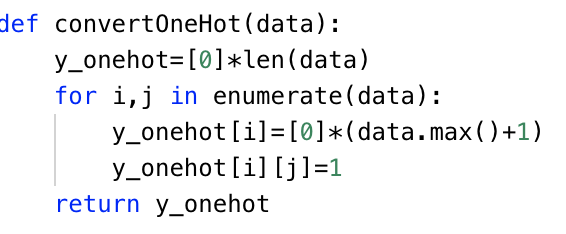


* Define the learning rate, number of epochs and batch size for artificial neural network

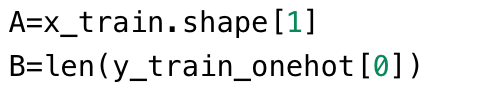


* An extra step in preprocess is to perform the one-hot encoding for the labels.

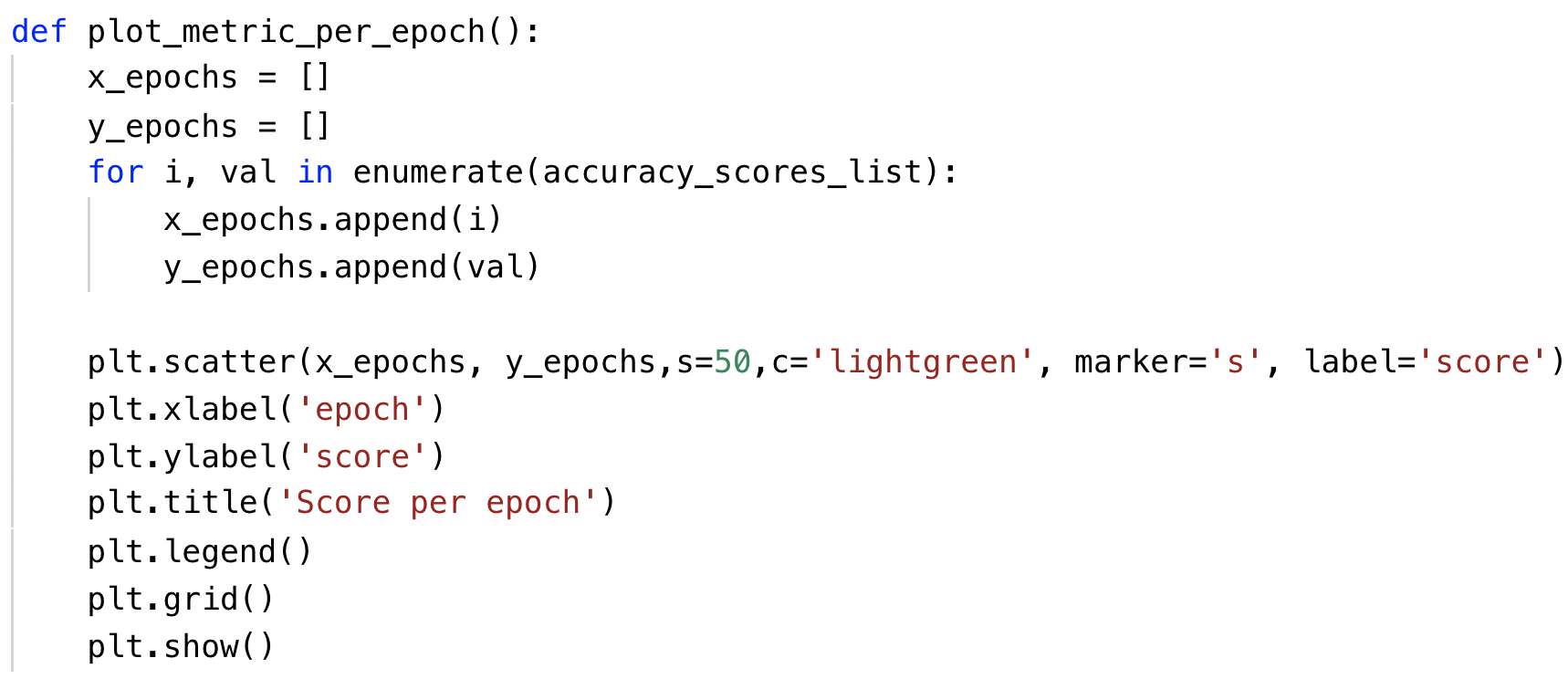




* Define the parameters to store the shape of placeholder.

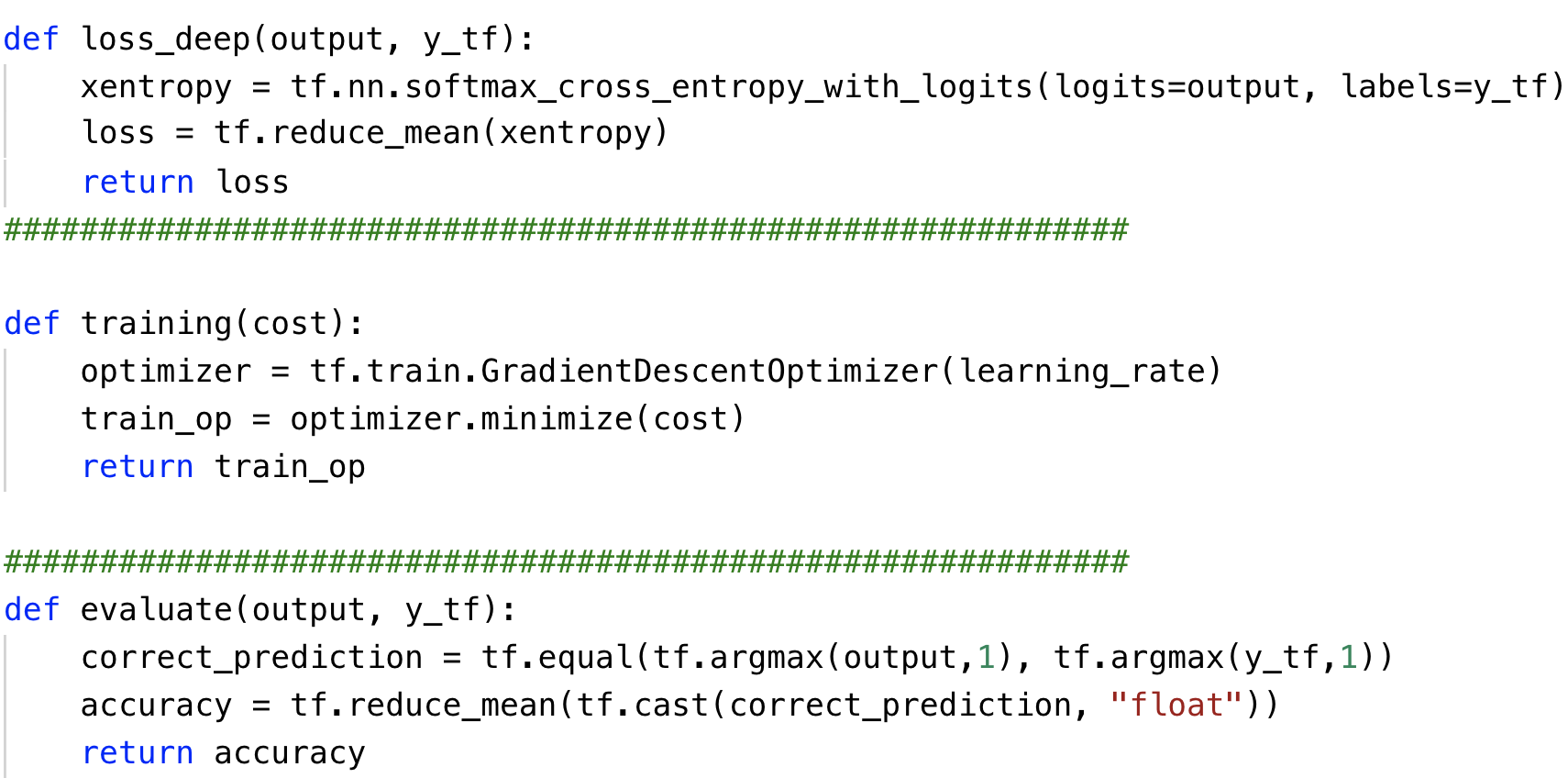


* Define the function to draw the plot of performance

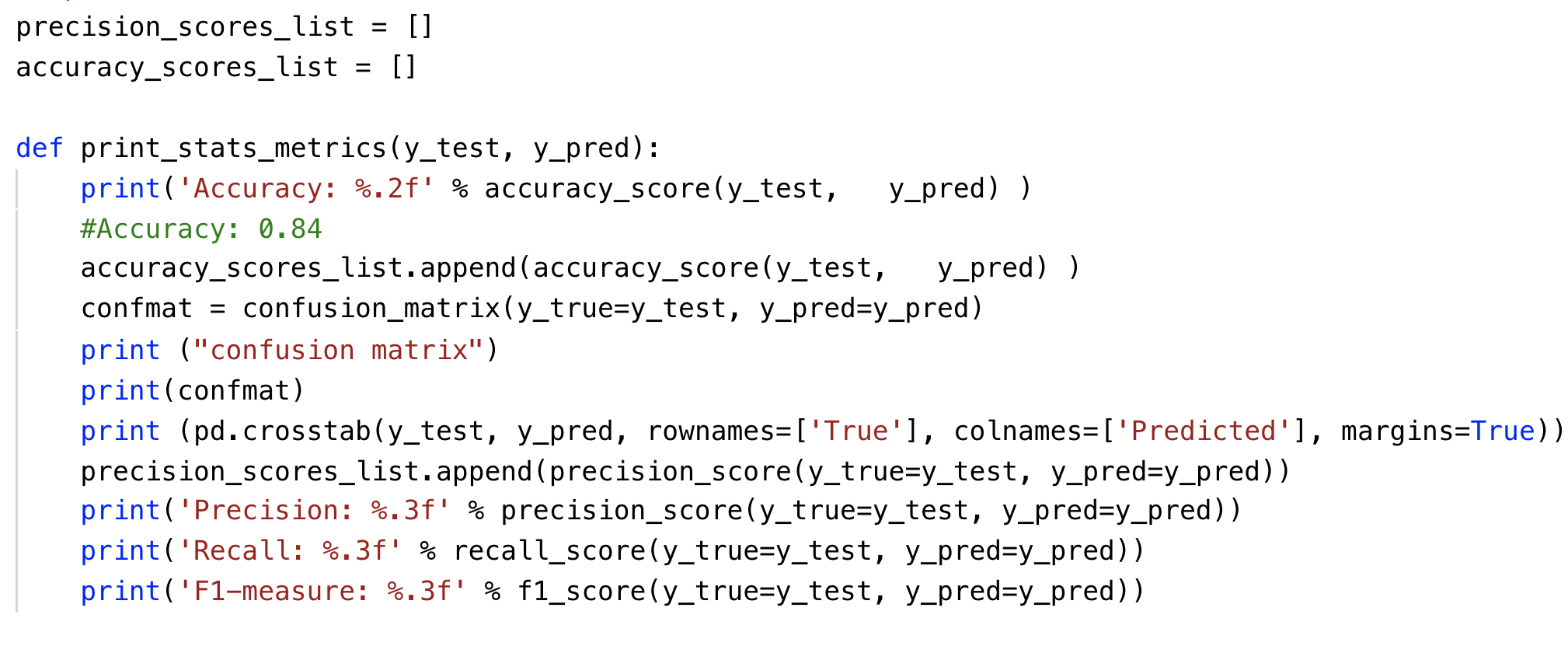


* Define your own architecture of neural network

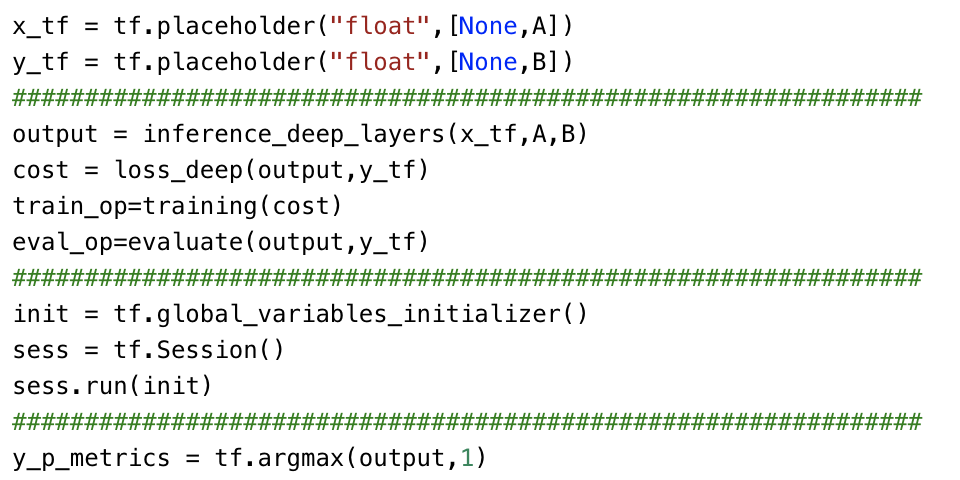




* Please print the statistics metrics such as accuracy, recall, precision and f1 score.



* Initialize the variables and placeholders. Then perform the training and testing on subset of kdd dataset.



## What to Submit

You should submit a lab report file which includes:

* + The steps for how you preprocessed data
  + The necessary code snippet of your classifier and architecture.
  + The screenshot of the results
  + You can name your report "Lab\_kdd\_small\_yourname.doc".