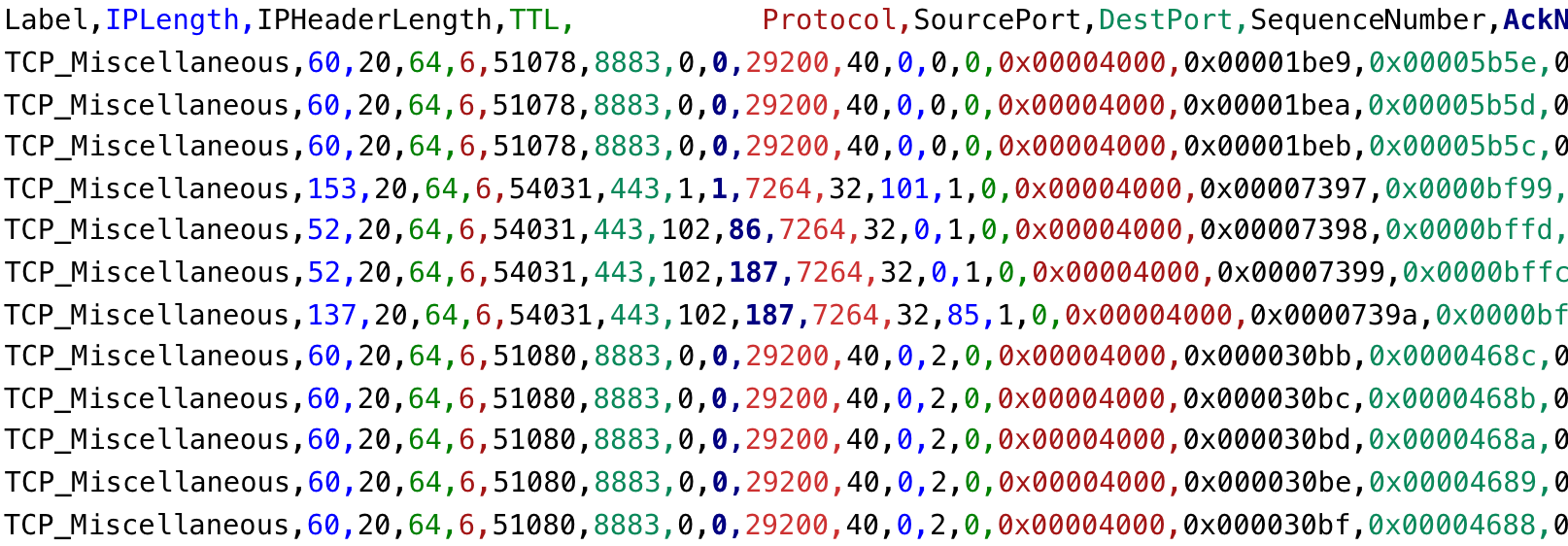
Machine learning cybersecurity

**Device detection**

# LAB 2: Writing a cLAssifier for the device detection

**Lab Description:** This lab is to classify the records of packet features to predict which device send the packet.

* The dataset contains 13138 records (miscellaneous:944, camera: 8010, assistant: 2178, outlet: 1802 and mobile: 186) with 18 features for each.



* The features and class values are both extracted from the pcap files.
* Tasks:
  + Use WEKA to input the dataset and perform the classification
  + Write a python script based on the sklearn library to implement the classifiers
  + Write a python script based on Tensorflow framework to implement the classifier.

**Lab Environment:**

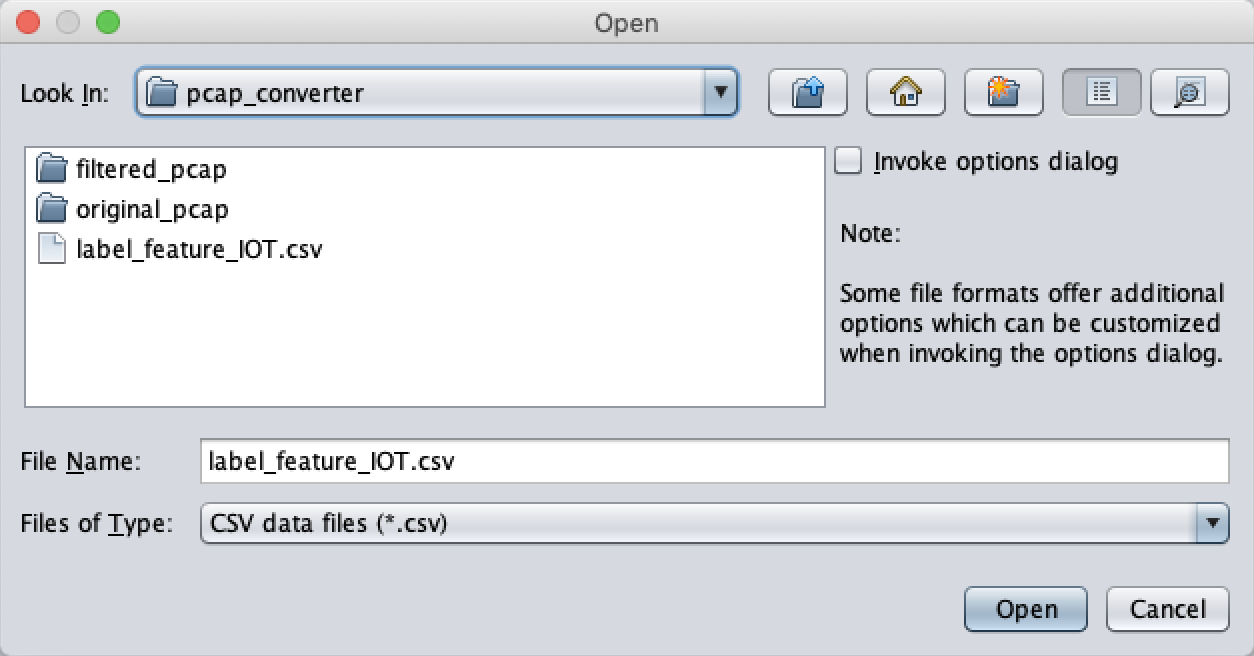
* The students should have access to a machine with Linux system
* WEKA should be installed
* The environment for python is required as well as some packages such as numpy, tensorflow and sklearn.

**Lab Files that are Needed:**

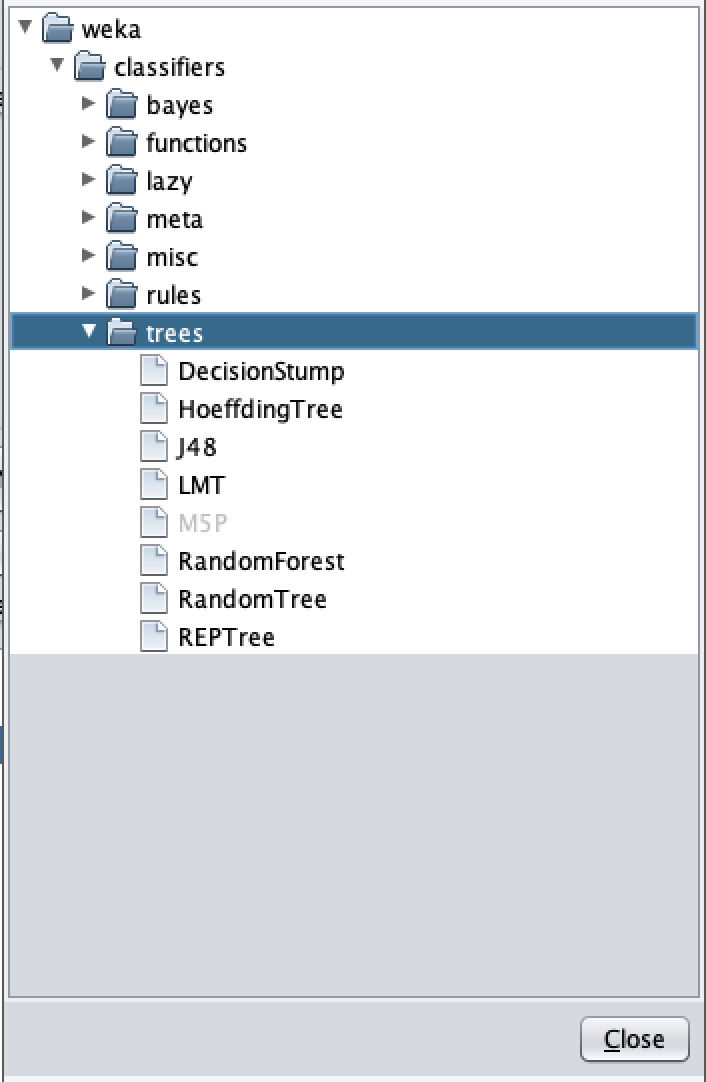
* For this lab you will need only one file (label\_feature\_IOT.csv) for both WEKA and python scripts.
* The first column is the class value, others are the features.

### **Lab exercise 1**

* Import data into WEKA (explorer), the files of type should be specified (csv).



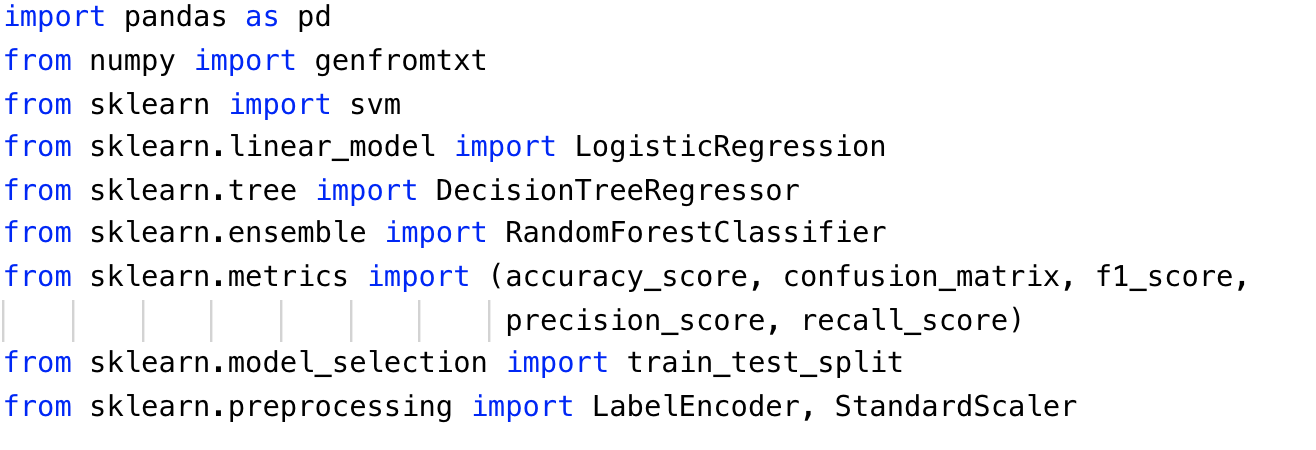
* Choose a proper classifier, such as RandomForest



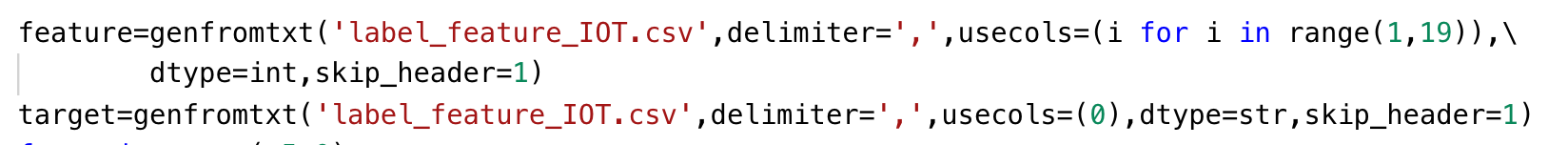
* Specify the test option and the column of class

### **Lab exercise 2**

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

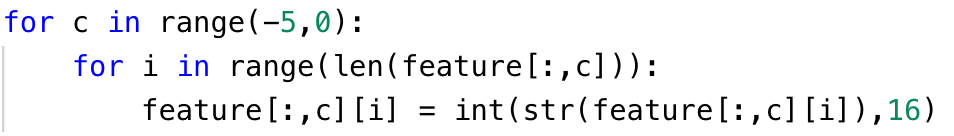


* Read the features and class values from malware dataset with proper method

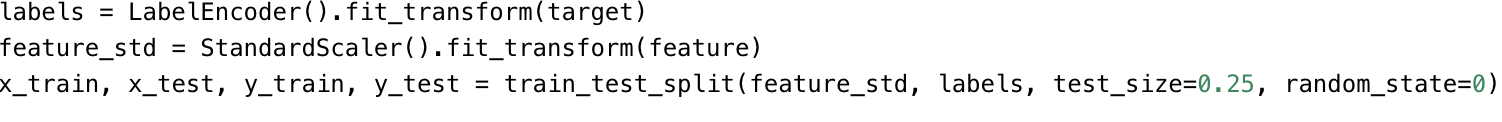


* Label\_feature\_IOT.csv is the name of the file.
* delimiter indicates the character to split the data in a row.
* usecols indicates which columns will be read. For features, the columns from 2 to 19 will be read. For class values, the first columns of the rows will be read.
* dtype indicates the type 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 last four columns of features are hexadecimal digits, they need to be converted to decimal digits.

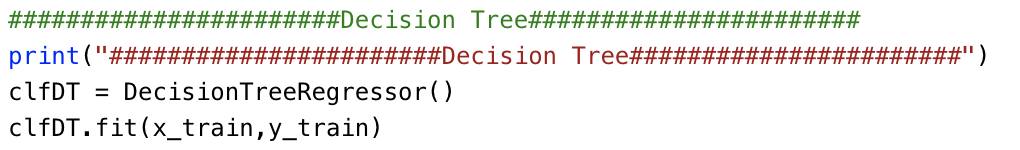
The following for loop will process each value in each column.



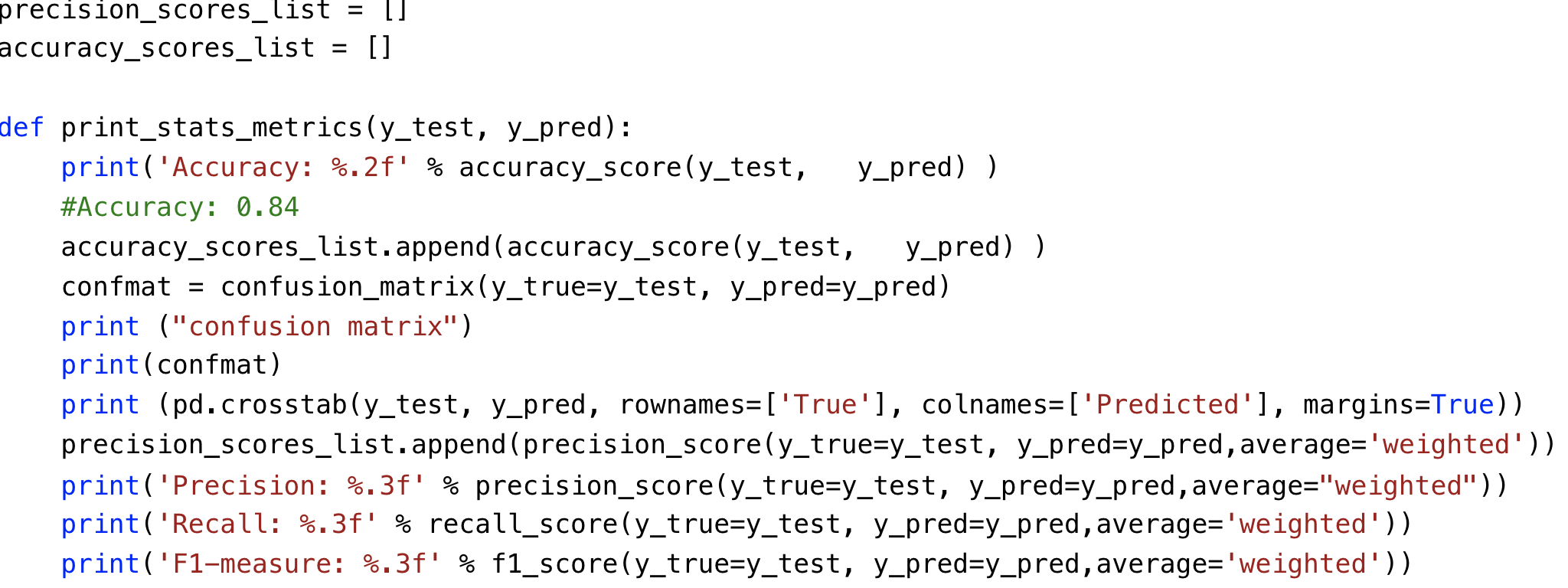
* Split the dataset. When you finish the preprocess step, you can write the python script with the use of sklearn package to build your architecture of classifier.



* random\_state is the seed used by the random number generator
* This is for the decision tree:



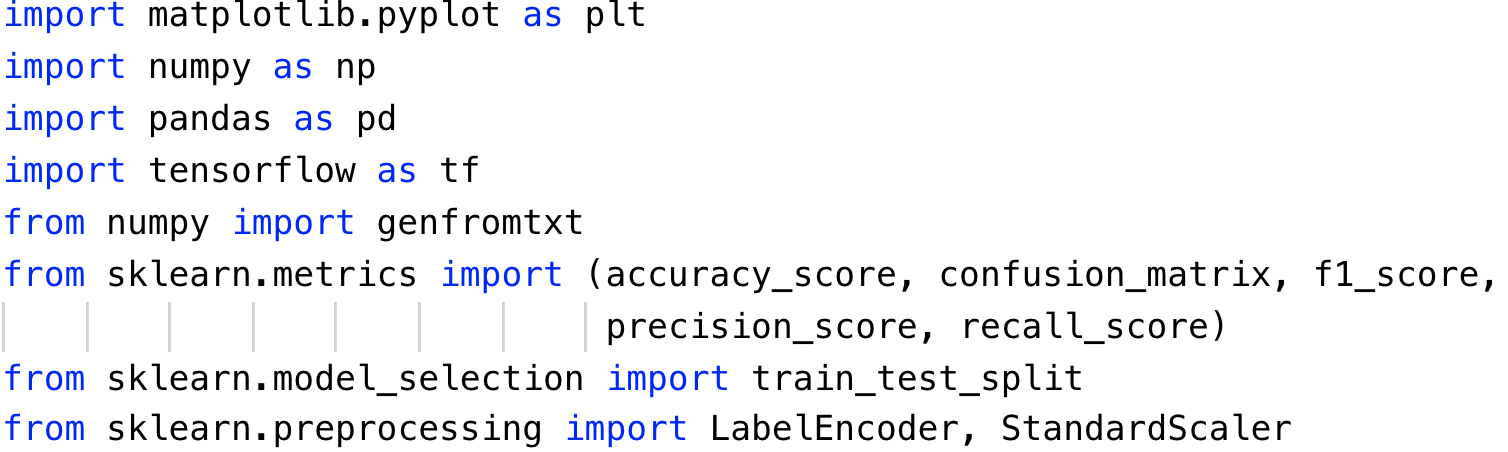
* Please print the statistics metrics such as accuracy, recall, precision and f1 score.



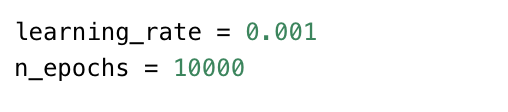
* Implement the classifiers based on Logistic Regression, Decision Tree, Naïve Bayes and Random Forest

### **Lab exercise 3**

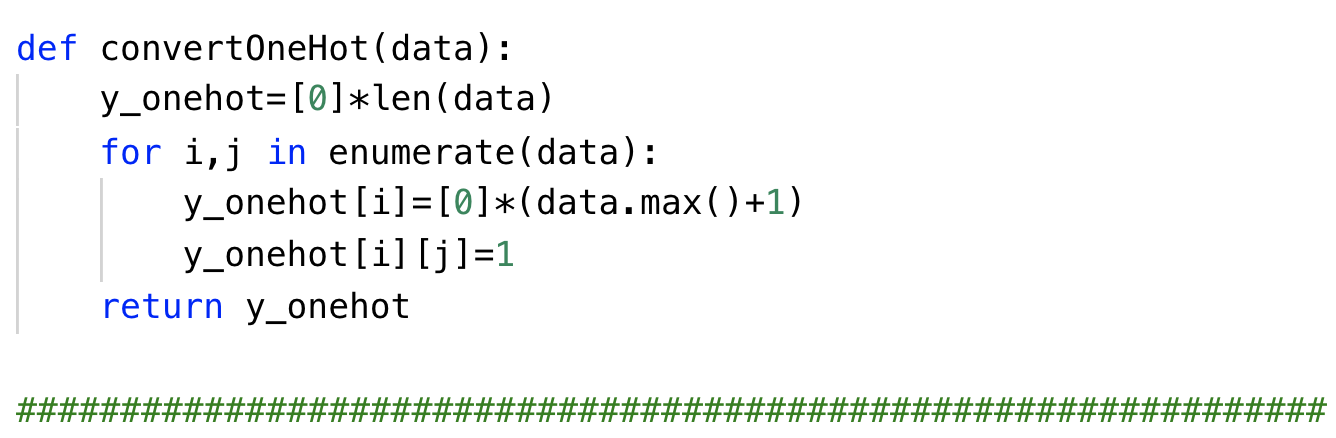
* Use the same data you use in the exercise 1 and 2.
* In this exercise, you will implement an artificial neural network classifier based on Tensorflow
* Import the required libraries



* 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 and number of epochs for artificial neural network

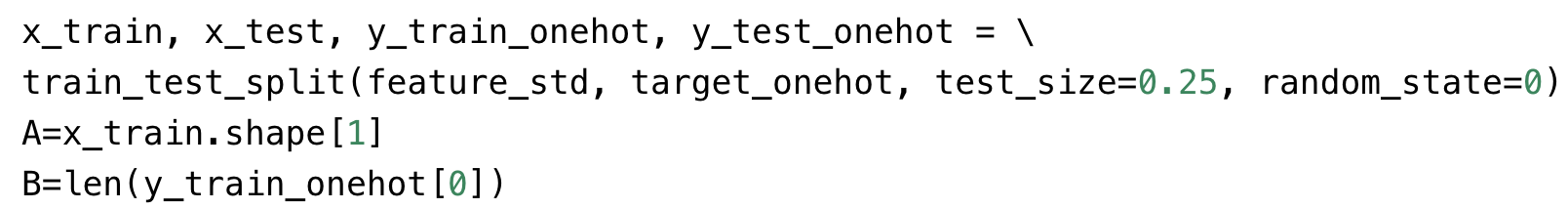


* An extra step in preprocess is to perform the one-hot encoding for the labels.





* Split the dataset after preprocessing and define the parameters to store the shape of placeholder.



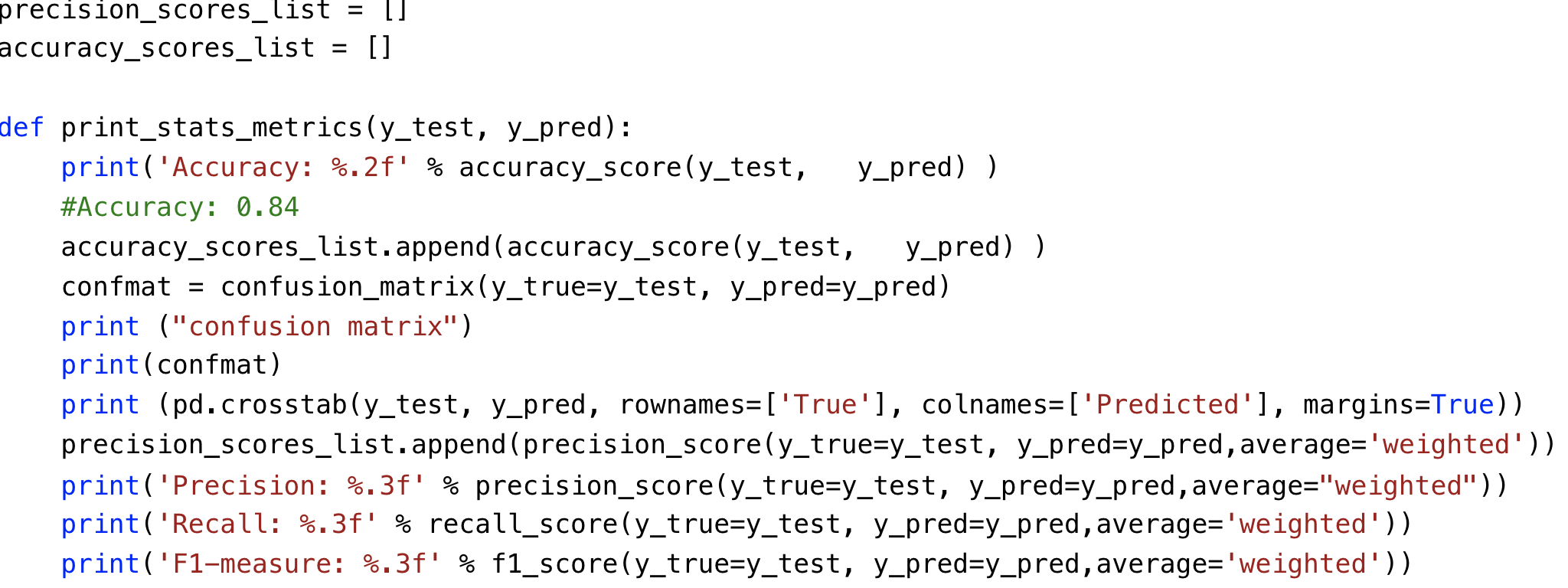
* Define the function to draw the plot of performance



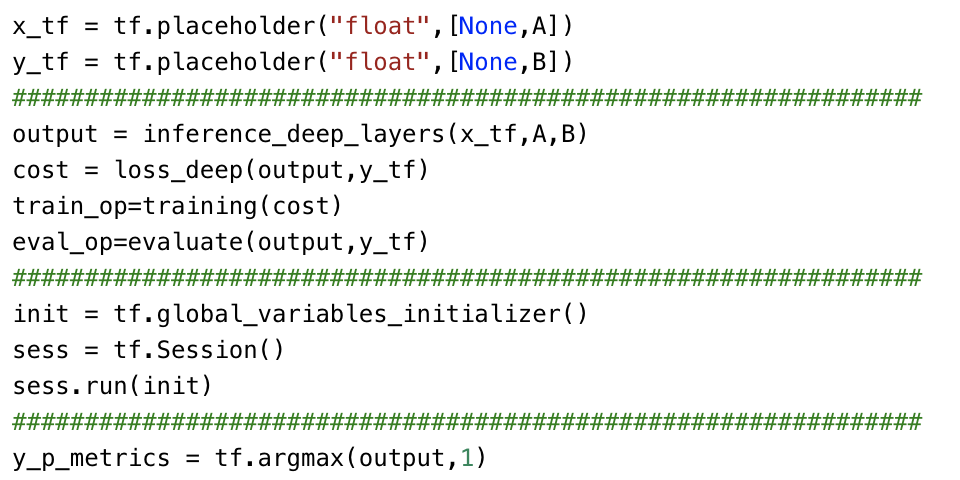
* Define your own architecture of neural network



* 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 iris dataset.



## What to Submit

You should submit a lab report file which include:

* The steps you preprocessed data
* The necessary code snippet of your classifier and architecture.
* The screenshot for both your code snippet and the result are needed.
* You can call your file "Lab\_IOT\_yourname.doc".