Machine learning cybersecurity **PHISHING detection**

# LAB 2: Writing a classifier for PHISHING DATASET

**Lab Description:** This lab is to write the python script as well as use WEKA to implement a binary classifier to estimate whether a website is a phishing website. The dataset contains 102816 web hits and 30 features were recorded for each of the hit. Also, a class value has been given for each of the record.

Example of phishing dataset:



Features Description:





You are required to implement it in three ways:

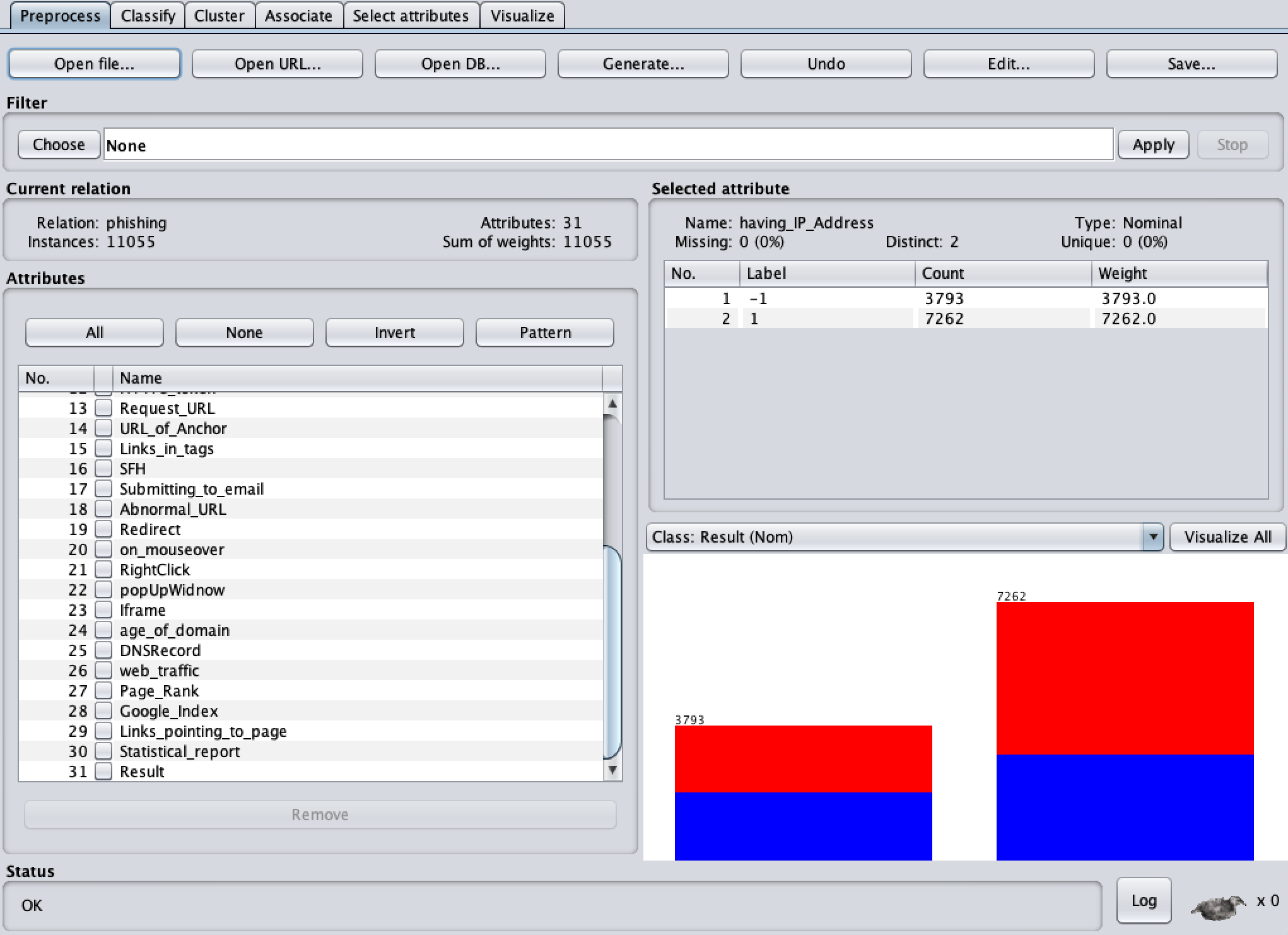
* Using the machine learning software WEKA.
* Writing a python script with the use of the package sklearn
* Writing a python script with the use of the package tensorflow and deep learning techniques.

**Lab Environment:** The student should have access to no matter a machine with Linux system or Windows system, but the environment for python is required as well as some packages such as numpy, tensorflow and sklearn.

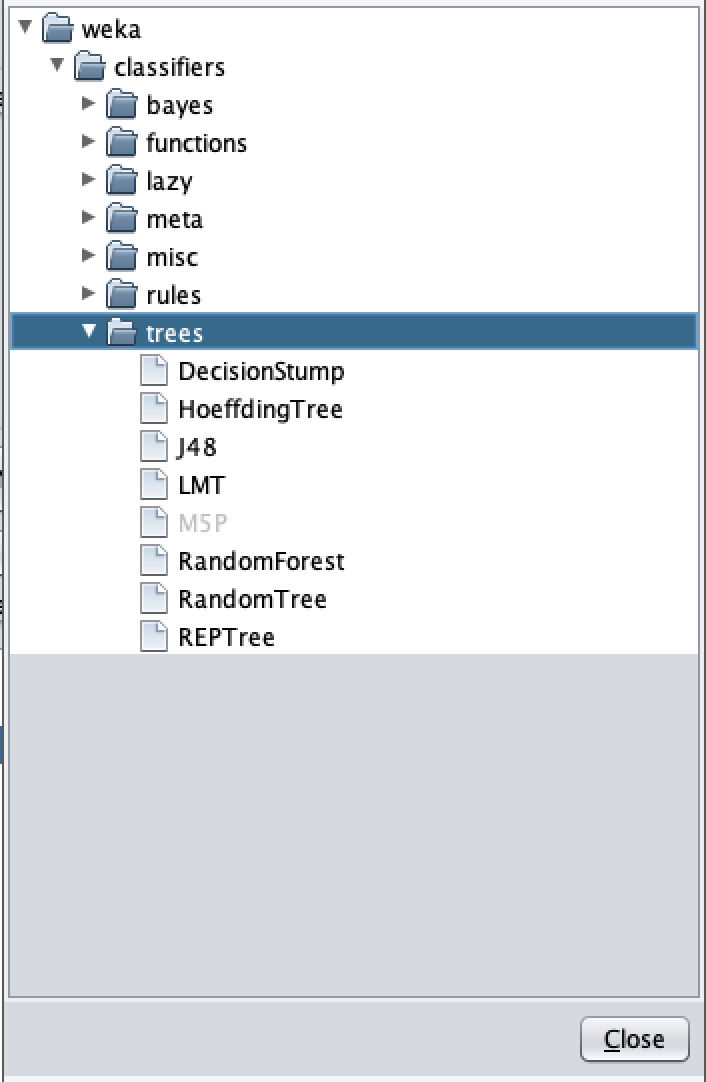
**Lab Files that are Needed:** For this lab you will need one file (phishing\_1.csv) the last column is the class value, others are the features.

### **Lab exercise 1**

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



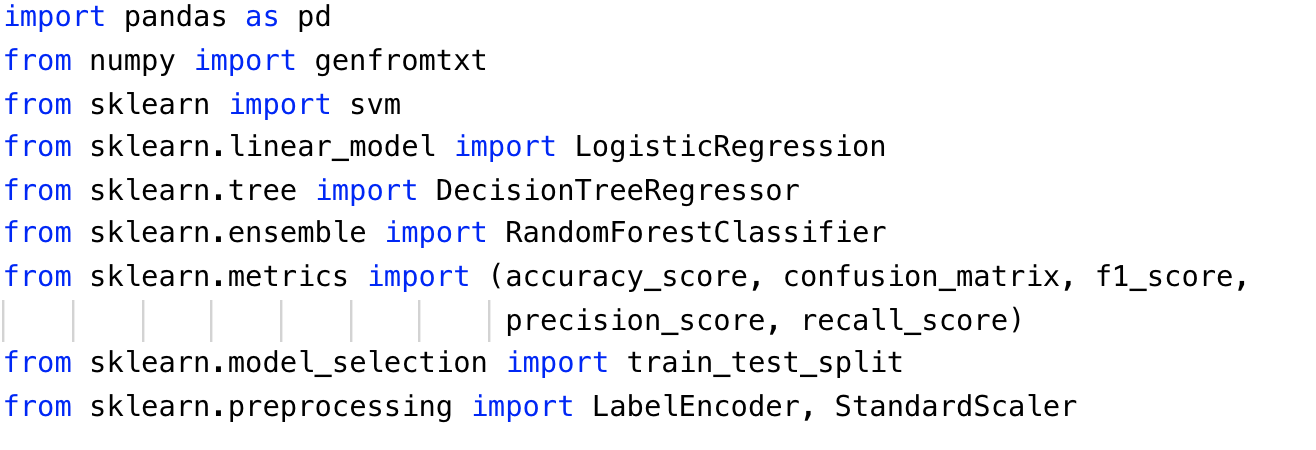
* Choose a proper classifier, such as RandomForest



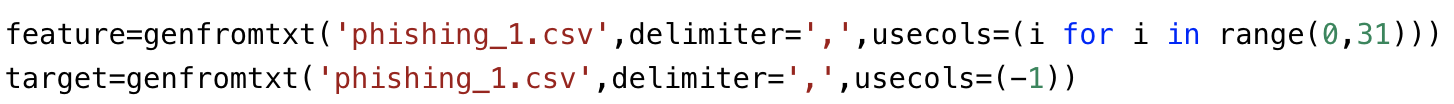
* Specify the test option and the column of class

### **Lab exercise 2**

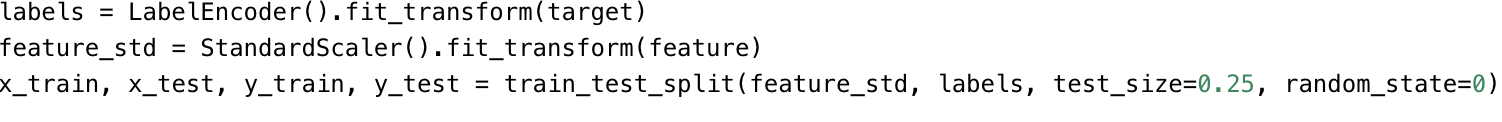
* In this exercise, you need to implement several classifiers with the use of sklearn.
* Import sklearn code and required libraries



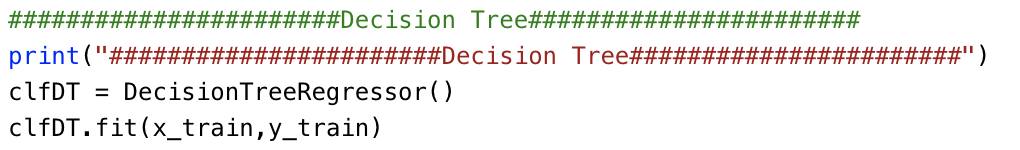
* Read the features and class values from malware dataset with proper method



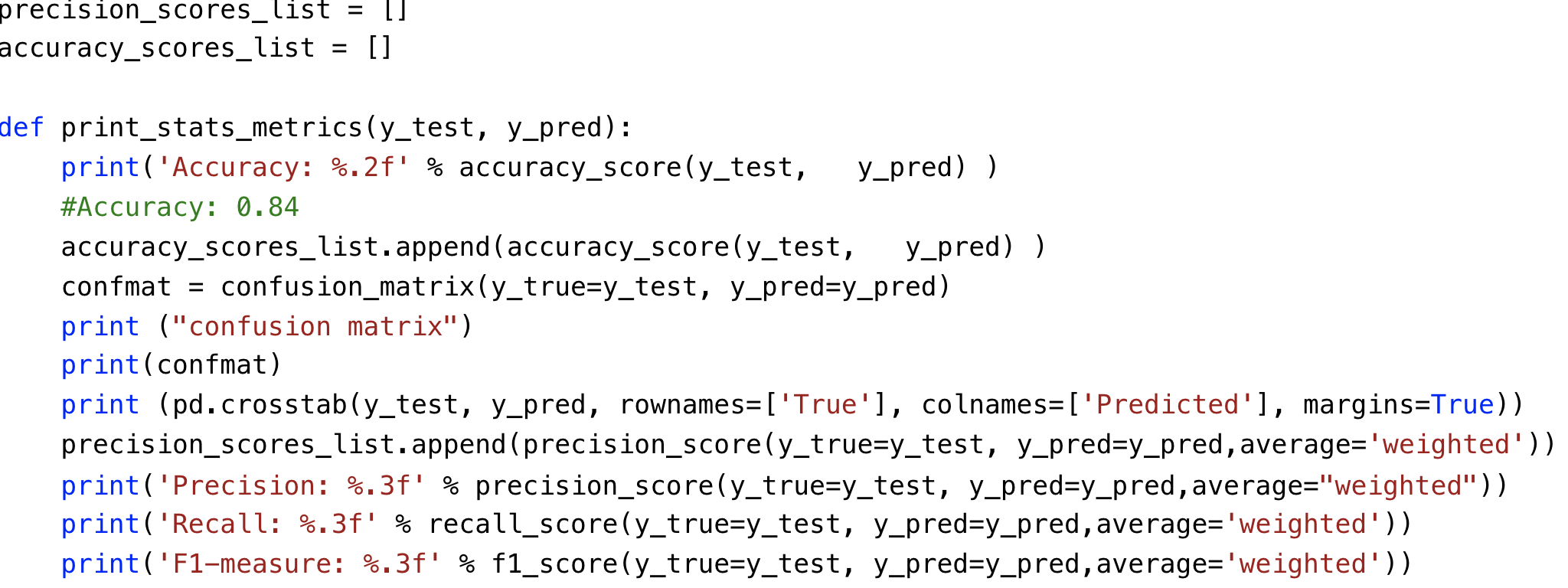
* Phishing\_1.csv is the name of the file.
* delimiter indicates the character to split the data in a row.
* usecols indicates which columns will be read. For features, the columns from 1 to 30 will be read. For class values, the first columns of the rows will be read.
* dtype indicates the type of data 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.
* 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:



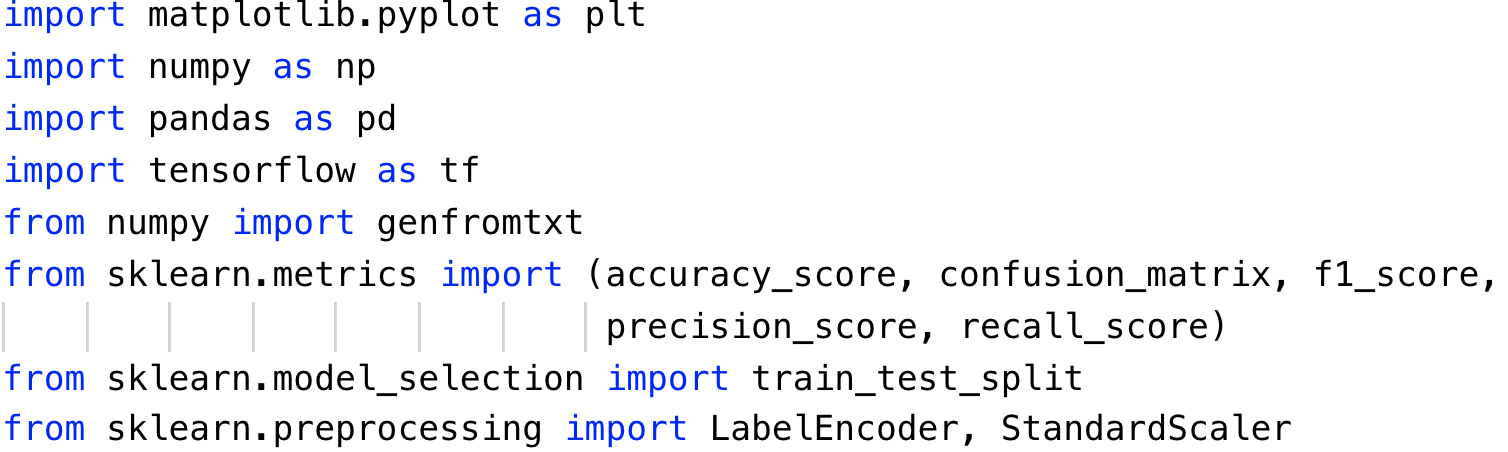
* Please print the statistics metrics such as accuracy, recall, precision and f1 score.



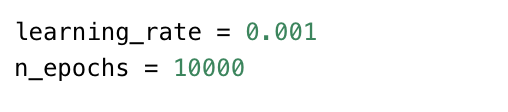
* Implement the classifiers based on Logistic Regression, Decision Tree, Naïve Bayes and Random Forest

### **Lab exercise 3**

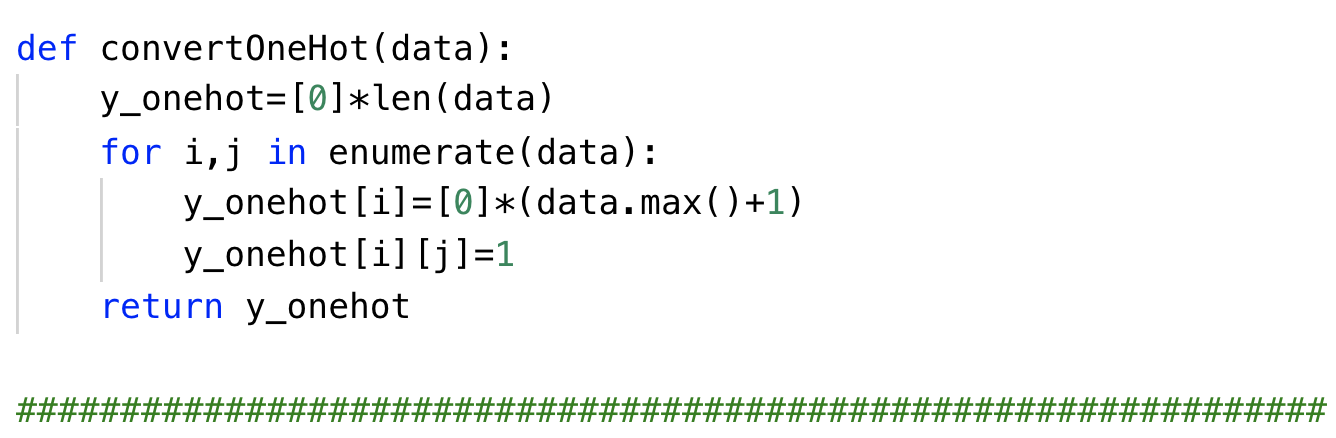
* Use the same data you use in the exercise 1 and 2.
* In this exercise, you will implement an artificial neural network classifier based on Tensorflow
* Import the required libraries

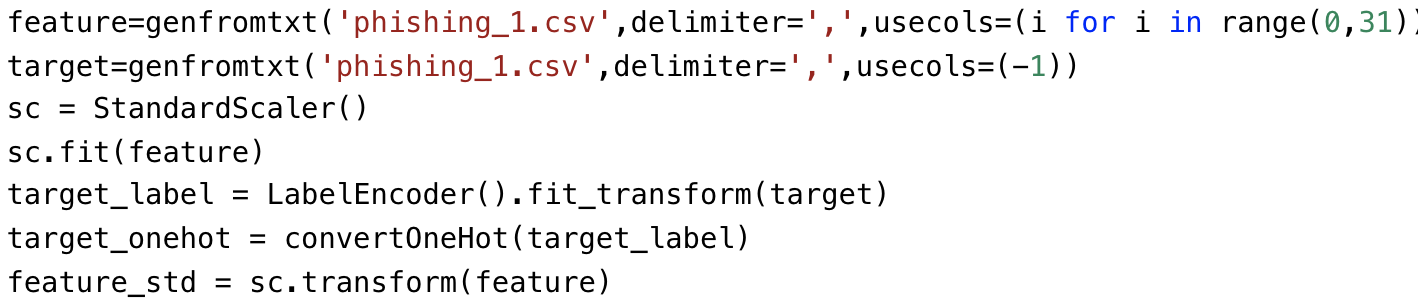


* Repeat the same steps to preprocess the data as Exercise 2. Read the data, standard scale the feature and encode the labels.
* Define the learning rate and number of epochs for artificial neural network

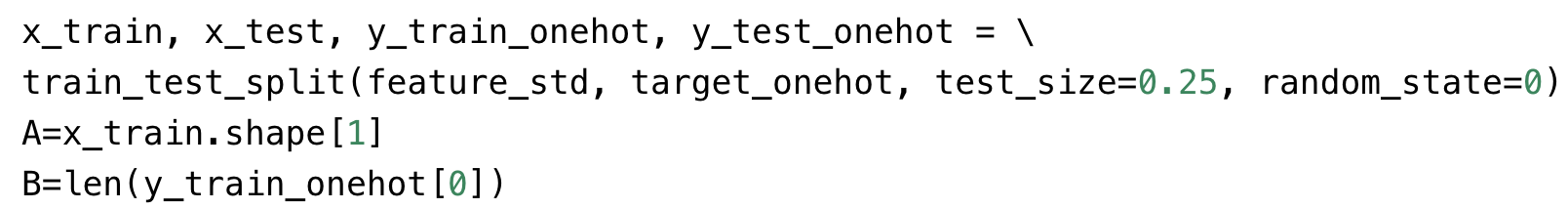


* An extra step in preprocess is to perform the one-hot encoding for the labels.





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



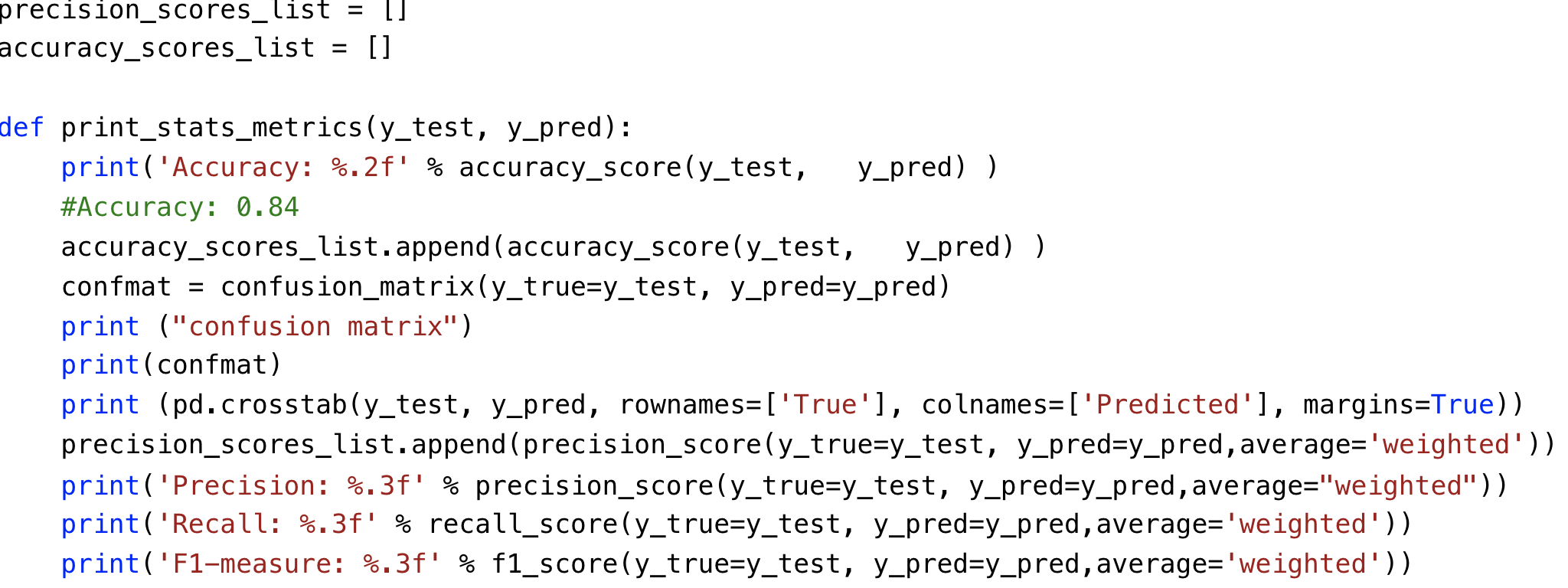
* Define the function to draw the plot of performance



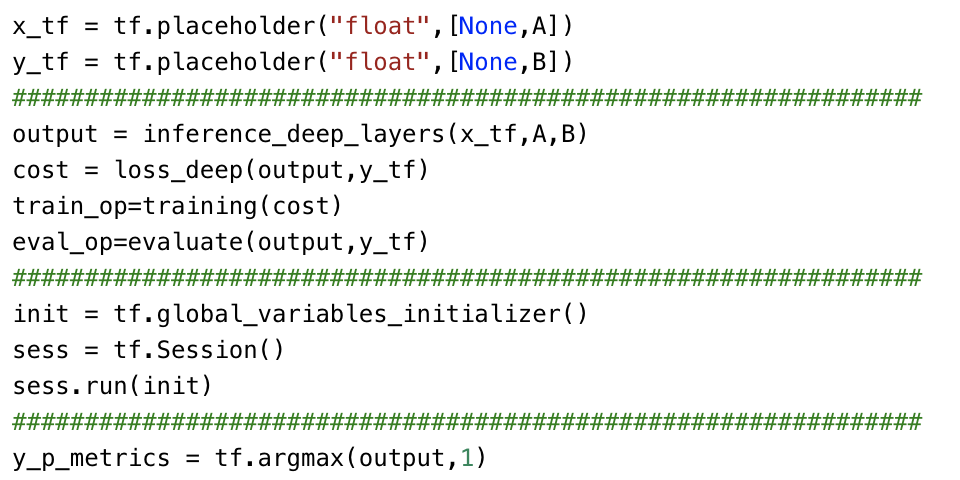
* Define your own architecture of neural network



* 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 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. Also, the screenshot for both your code snippet and the result are needed. You can call your file "Lab2\_phishing\_yourname.doc".