National College of Ireland

Configuration Manual

MSc Internship

Cyber Security

Arun Kollara

Student ID: 18212204

School of Computing

National College of Ireland

Supervisor: Ross Spelman

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Arun Manoharan Kollara		
Student ID:	18212204		
Programme:	MSc in Cybersecurity	**Year:**	2019/2020
Module:	Academic Internship		
Lecturer:	Prof. Ross Spelman		
Submission Due Date:	17/08/2020		
Project Title:	Opcode Frequency Based Malware Detection Using Hybrid Classifiers		
Word Count:	334	**Page Count:**	7

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.
I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

Signature: Arun Manoharan Kollara

Date: 17/08/2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Arun Kollara
Student ID: 18212204

1 Introduction

This document specifies the system requirements and configuration details for the "Opcode Frequency Based Malware Detection Using Hybrid Classifiers" research project.

2 System Requirements

The entire code is developed in Python version 2 on Ubuntu 14.04, using Pipenv based virtual environment. In order to replicate the project on any system, it is highly recommended to use the same version of Python and Operating System (Ubuntu), as well as other dependent libraries as discussed below. Also, all python dependent libraries are automatically installed from the Pipfile (included in the project) as per the version number mentioned.

2.1 System packages needed to be installed are:

i. python 2.7
ii. python-tk
iii. python-pip

2.2 Python packages needed to be installed are:

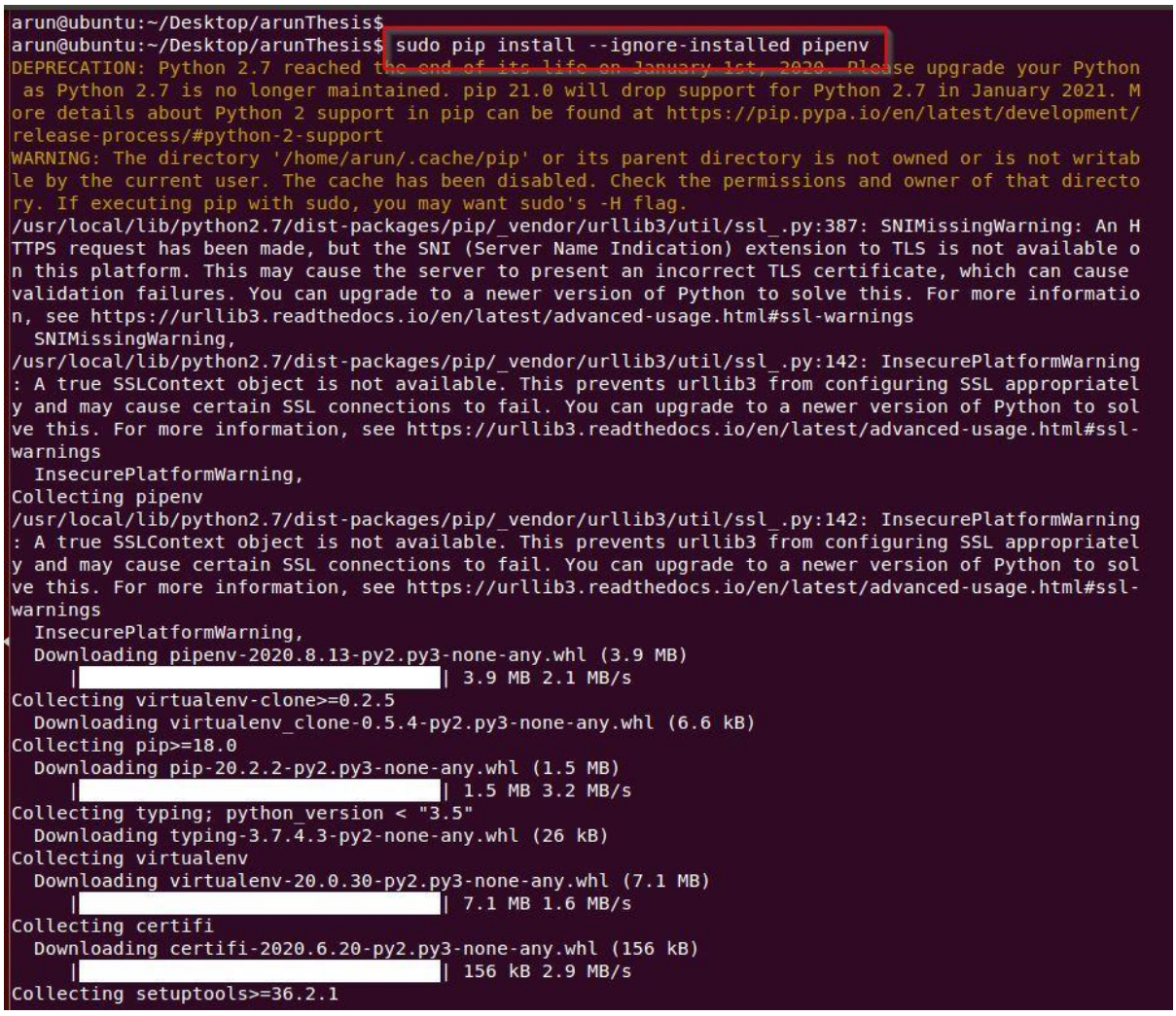
i. pipenv
ii. scikit-learn==0.20.0
iii. xgboost==0.82
iv. pandas==0.24.2
v. matplotlib==2.2.5
vi. seaborn==0.9.1

2.3 For setup, run following commands in order:

i. sudo apt-get install -y python python-pip python-tk

```
arun@ubuntu:~/Desktop/arunThesis$
arun@ubuntu:~/Desktop/arunThesis$ sudo apt-get install -y python python-pip python-tk
Reading package lists... Done
Building dependency tree
Reading state information... Done
python is already the newest version.
python set to manually installed.
python-tk is already the newest version.
python-pip is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
arun@ubuntu:~/Desktop/arunThesis$
```

ii. sudo pip install --ignore-installed pipenv



iii. cd /_path_/_to_/_projectFolder_/
iv. pipenv install --skip-lock

```
arun@ubuntu:~/Desktop/arunThesis$
arun@ubuntu:~/Desktop/arunThesis$ ls -l
total 1892
-rw-rw-r-- 1 arun arun  134015 Aug 16 02:12 Bening.csv
-rw-rw-r-- 1 arun arun    1473 Aug 16 21:38 check.csv
-rw-rw-r-- 1 arun arun    3800 Aug 15 13:28 classification_rpt.py
-rw------- 1 arun arun     213 Aug 15 15:42 configure.sh
drwxrwxr-x 5 arun arun    4096 Aug 15 16:37 data
-rw-rw-r-- 1 arun arun 1601482 Aug 16 21:38 hello.txt
-rw-rw-r-- 1 arun arun  156003 Aug  9 22:20 Malware.csv
-rw-rw-r-- 1 arun arun    1080 Aug 15 13:26 opcode_dataset.py
-rw-rw-r-- 1 arun arun    1367 Aug 15 16:41 opcode_generate.py
-rw-rw-r-- 1 arun arun    3788 Aug 16 04:25 output.txt
-rw-rw-r-- 1 arun arun     277 Aug 15 15:47 Pipfile
-rwxrw-rw- 1 arun arun    7614 Aug 16 02:04 project.py
arun@ubuntu:~/Desktop/arunThesis$
arun@ubuntu:~/Desktop/arunThesis$ cat Pipfile
[[source]]
name = "pypi"
url = "https://pypi.org/simple"
verify_ssl = true

[dev-packages]

[packages]
scikit-learn = "==0.20.0"
pandas = "==0.24.2"
matplotlib = "==2.2.5"
seaborn = "==0.9.1"
xgboost = "==0.82"
ipython = "*"

[requires]
python_version = "2.7"
arun@ubuntu:~/Desktop/arunThesis$
```

```
arun@ubuntu:~/Desktop/arunThesis$ pipenv install --skip-lock
Creating a virtualenv for this project...
Pipfile: /home/arun/Desktop/arunThesis/Pipfile
Using /usr/bin/python2.7 (2.7.6) to create virtualenv...
⠸ Creating virtual environment...created virtual environment CPython2.7.6.final.0-64 in 135ms
  creator CPython2Posix(dest=/home/arun/.local/share/virtualenvs/arunThesis-62SbDRzL, clear=Fal
  seeder FromAppData(download=False, pip=bundle, wheel=bundle, setuptools=bundle, via=copy, app
share/virtualenv)
    added seed packages: pip==20.2.1, setuptools==44.1.1, wheel==0.34.2
  activators PythonActivator,CShellActivator,FishActivator,PowerShellActivator,BashActivator

✔ Successfully created virtual environment!
Virtualenv location: /home/arun/.local/share/virtualenvs/arunThesis-62SbDRzL
Installing dependencies from Pipfile...
  01F 400 ████████████████████████████████ 6/6 — 00:00:27
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
arun@ubuntu:~/Desktop/arunThesis$
```

3 Data Sources

The size of the dataset is about 3000 executable files. 2000 legitimate files were obtained from online free software sources like SourceForge, PortableApps and Softsonic. 1000 malware samples were downloaded from Virusshare. All the files in the dataset are 32bit.

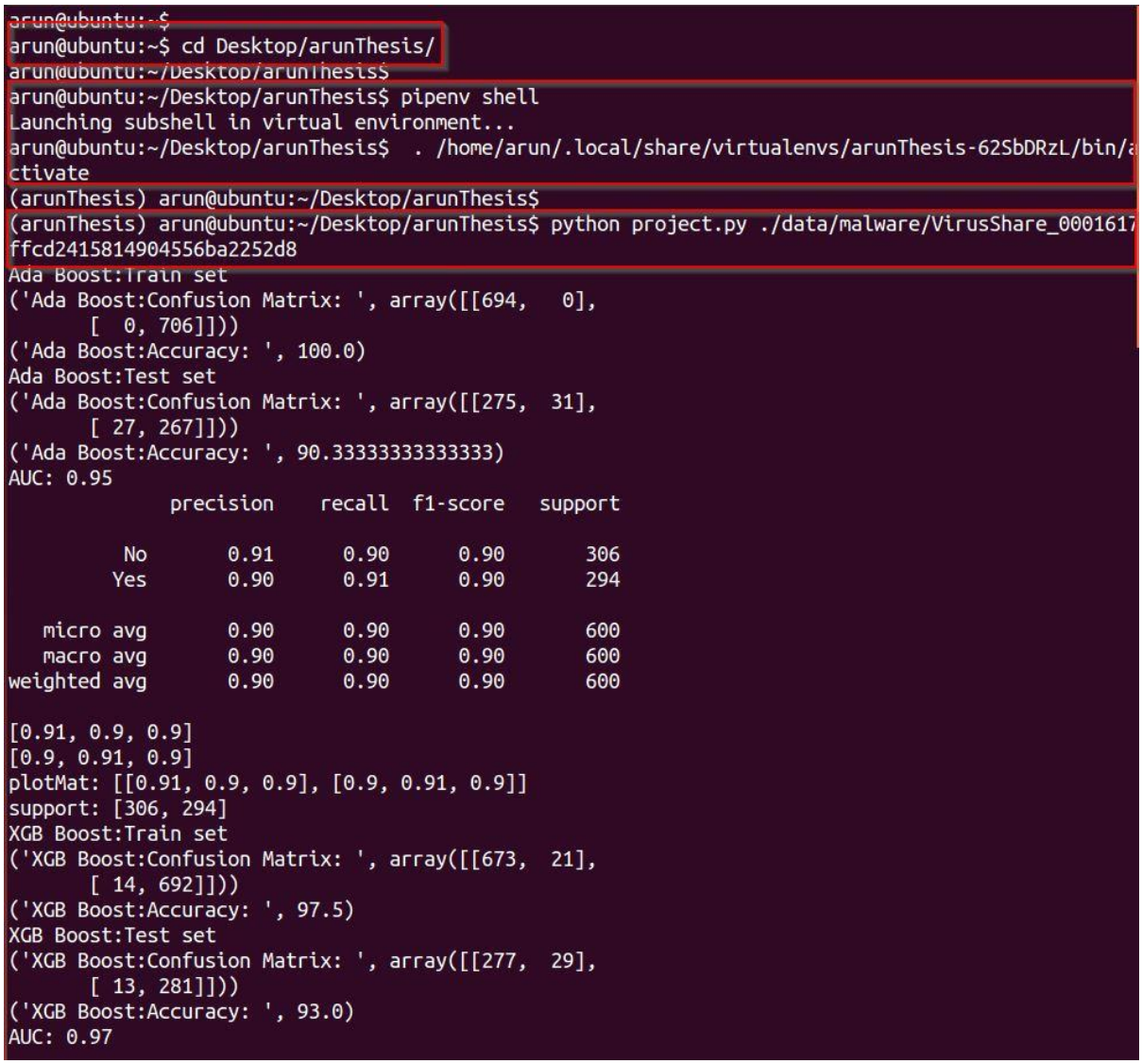
4 Code Execution

Open the terminal and navigate into the directory with the project code. Now, activate the virtual environment with the following command:

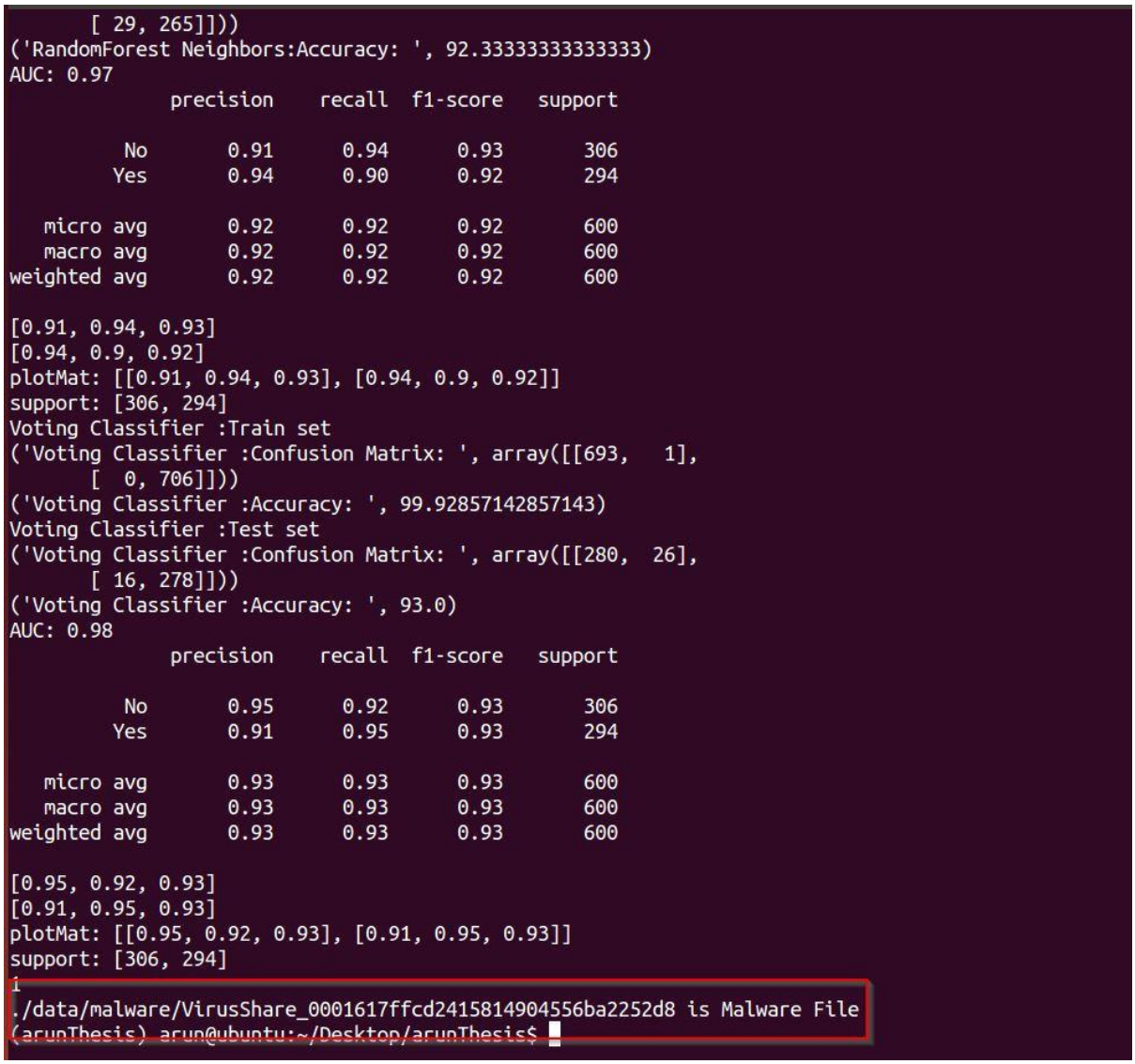
- pipenv shell

Now run the project with the following command:

- python project.py



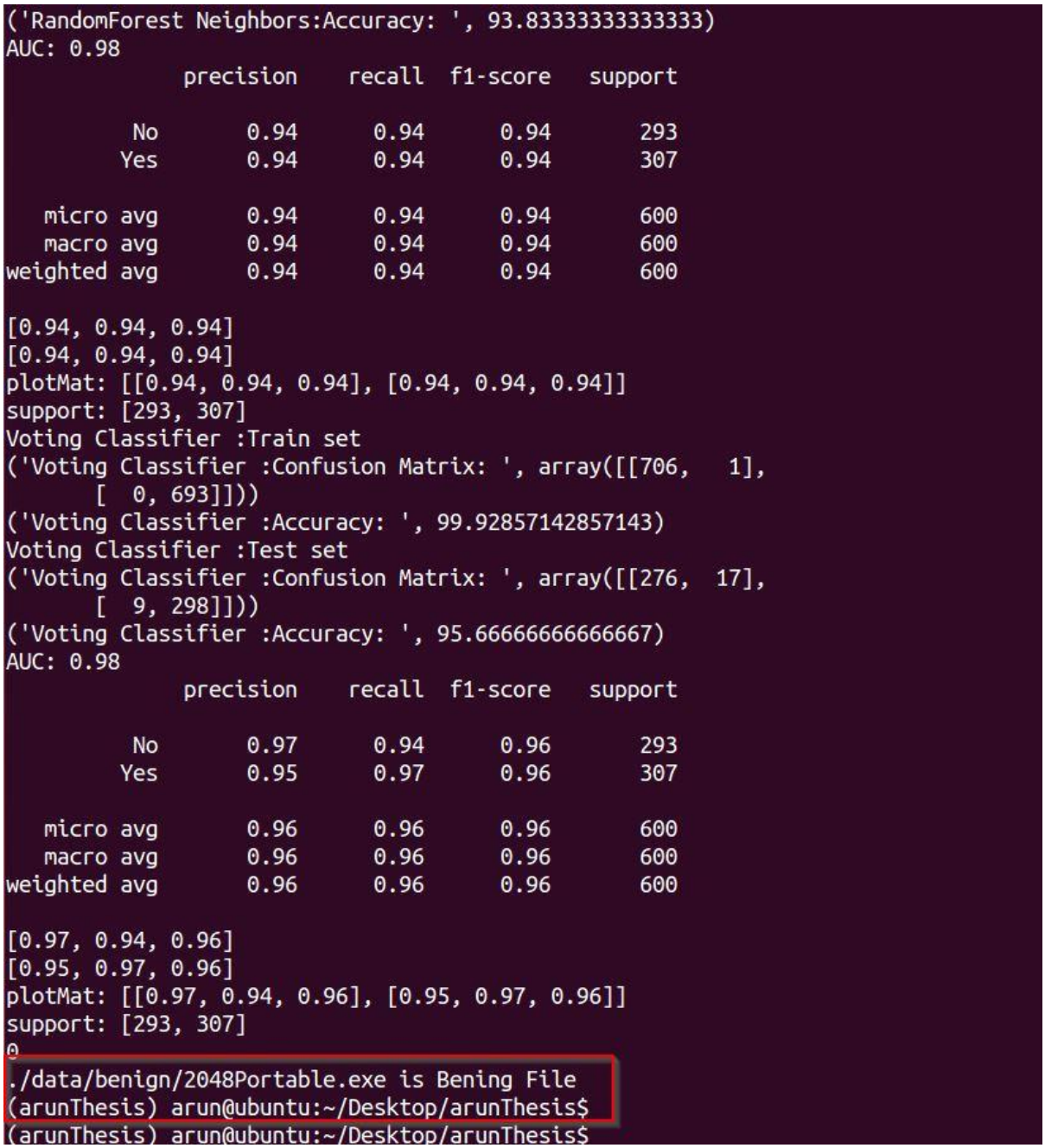
Above figure demonstrates the run of project to verify that the project is able identify a malicious file.



Above figure demonstrates the output of the run, as we can see that the project is able identify the malicious file.

```
(arunThesis) arun@ubuntu:~/Desktop/arunThesis$
(arunThesis) arun@ubuntu:~/Desktop/arunThesis$ python project.py ./data/benign/2048Portable.exe
Ada Boost:Train set
('Ada Boost:Confusion Matrix: ', array([[706,   1],
       [  0, 693]]))
('Ada Boost:Accuracy: ', 99.92857142857143)
Ada Boost:Test set
('Ada Boost:Confusion Matrix: ', array([[266,  27],
       [ 24, 283]]))
('Ada Boost:Accuracy: ', 91.5)
AUC: 0.96
             precision    recall  f1-score   support

         No       0.92      0.91      0.91       293
        Yes       0.91      0.92      0.92       307

   micro avg       0.92      0.92      0.92       600
   macro avg       0.92      0.91      0.91       600
weighted avg       0.92      0.92      0.91       600

[0.92, 0.91, 0.91]
[0.91, 0.92, 0.92]
plotMat: [[0.92, 0.91, 0.91], [0.91, 0.92, 0.92]]
support: [293, 307]
XGB Boost:Train set
('XGB Boost:Confusion Matrix: ', array([[682,  25],
       [ 12, 681]]))
('XGB Boost:Accuracy: ', 97.35714285714285)
XGB Boost:Test set
('XGB Boost:Confusion Matrix: ', array([[269,  24],
       [ 18, 289]]))
('XGB Boost:Accuracy: ', 93.0)
AUC: 0.98
             precision    recall  f1-score   support

         No       0.94      0.92      0.93       293
        Yes       0.92      0.94      0.93       307

   micro avg       0.93      0.93      0.93       600
   macro avg       0.93      0.93      0.93       600
```

Above figure demonstrates the run of project to verify that the project is able identify a benign file.



Above figure demonstrates the output of the run, as we can see that the project is able identify the benign file.