

Configuration Manual

MSc Internship Project
Cyber Security

Sudha Koride
Student ID: 20196083

School of Computing
National College of Ireland

Supervisor: Mr. Vikas Sahni.

National College of Ireland
MSc Project Submission Sheet
School of Computing



Student Name: Sudha Gaurinath Koride

Student ID: 20196083

Programme: M.Sc., Cyber Security

Year: Jan 2021-Jan2022.

Module: Research Internship Project

Lecturer: Mr. Vikas Sahni

Submission Due Date: 7th Jan 2022

Project Title: Fully Qualified Domain Name(FQDN) as Indexing Parameter for deduplicating the data in Network Devices.

Page count: 13

Word Count: 1169

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.

Signature:

A handwritten signature in blue ink, appearing to read "Sudha", on a light blue rectangular background.

Date:

7th Jan 2022.

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	<input type="checkbox"/>
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.	<input type="checkbox"/>

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

Sudha Koride
20196083

1 Introduction

The Configuration Manual contains the details of specifications of tools, hardware & software needed for implementation and execution of the Research Project. It also describes the stepwise procedure of executing the code to see the implementation and results obtained. Also, the monthly tasks that were carried out during the Internship period (Sept2021-Dec 2021) are mentioned in brief.

2 Environmental Setup

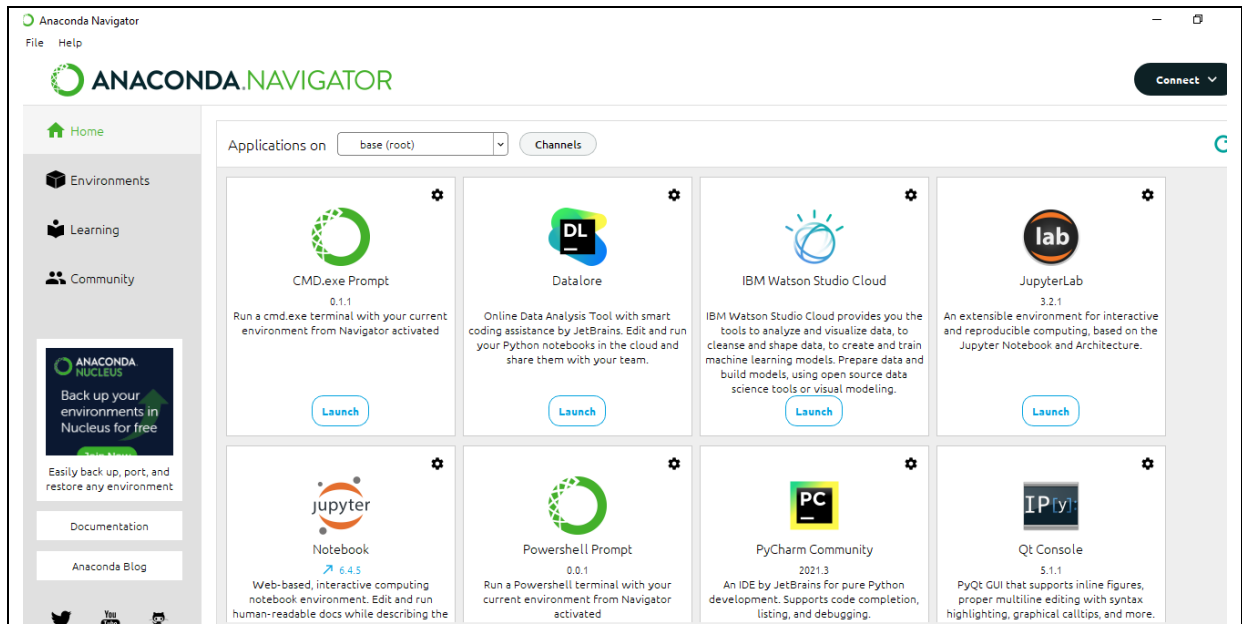
The proposed solution is implemented with below specification and configuration of Hardware and software:

- Operating System: Windows 10
- Operating System Type: 64-bit operating system, x64-based processor.
- Processor: Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz
- Memory: 12.0 GB
- Programming language: Python 3.7
- Environment: Jupyter Notebook 3

3 Installation of Tools

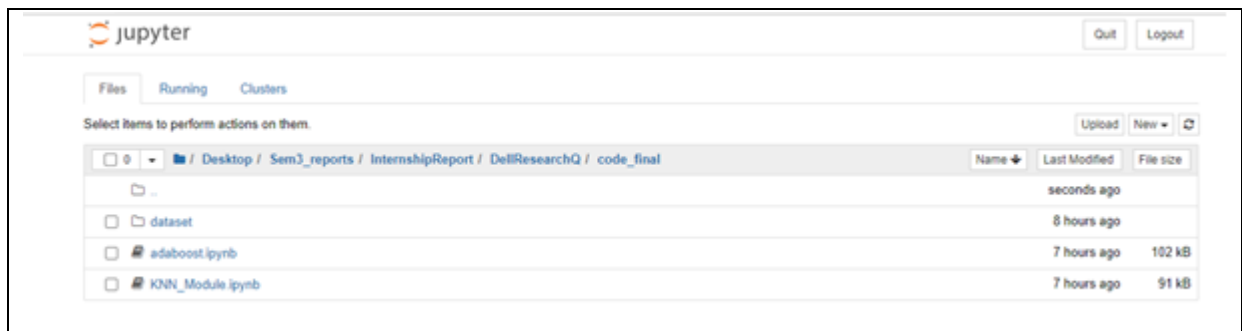
3.1 Anaconda Navigator 3

The Anaconda Navigator is a graphical user interface provided by Anaconda. It has many in built applications which can be easily launched and used with this tool.

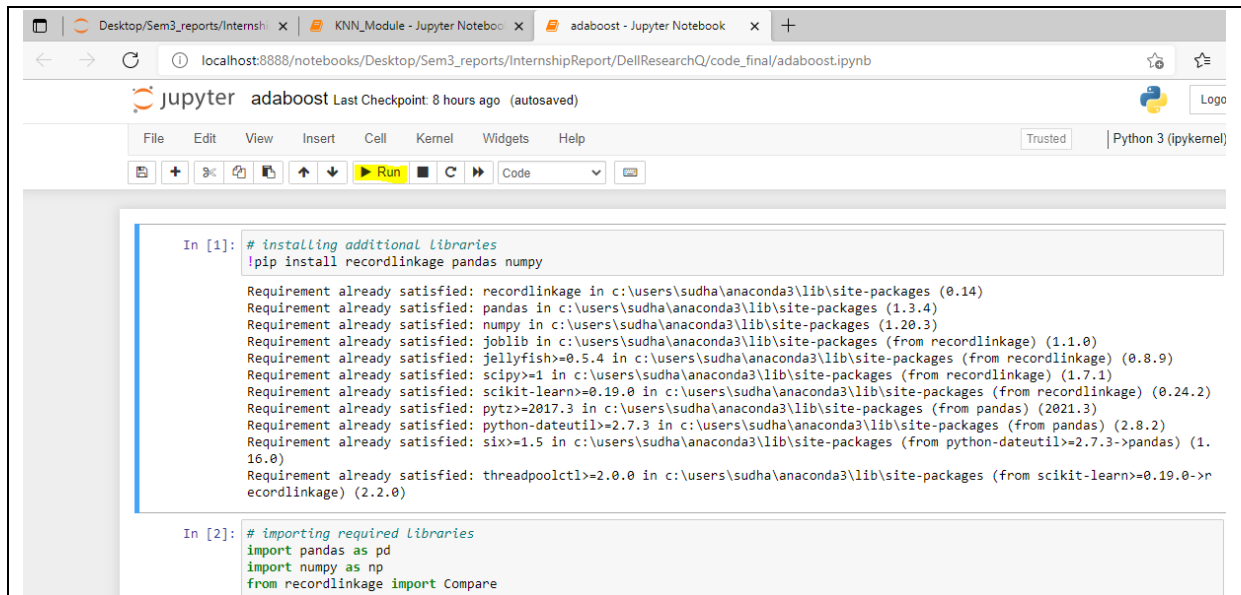


3.2 Jupyter Notebook 3:

After installing the Anaconda package, we can launch the Jupyter Notebook through search bar and then navigate to the location of the folder where the code that we want to run is located.



After that click on the program to execute. It will get uploaded in Jupyter Notebook as shown in the image below. Then click on “Run” to execute the code and check the output.



The screenshot shows a Jupyter Notebook window with two tabs: 'KNN_Module - Jupyter Notebook' and 'adaboost - Jupyter Notebook'. The active tab is 'adaboost - Jupyter Notebook'. The notebook is running on a local server at 'localhost:8888/notebooks/Desktop/Sem3_reports/InternshipReport/DellResearchQ/code_final/adaboost.ipynb'. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for saving, running, and other actions. The notebook content shows two code cells. Cell [1] contains a comment '# installing additional libraries' followed by the command '!pip install recordlinkage pandas numpy'. The output of this cell lists various requirements that are already satisfied, including recordlinkage (0.14), pandas (1.3.4), numpy (1.20.3), joblib (1.1.0), jellyfish (0.8.9), scipy (1.7.1), scikit-learn (0.24.2), python-dateutil (2021.3), six (2.8.2), and threadpoolctl (2.2.0). Cell [2] contains a comment '# importing required libraries' followed by the imports: 'import pandas as pd', 'import numpy as np', and 'from recordlinkage import Compare'.

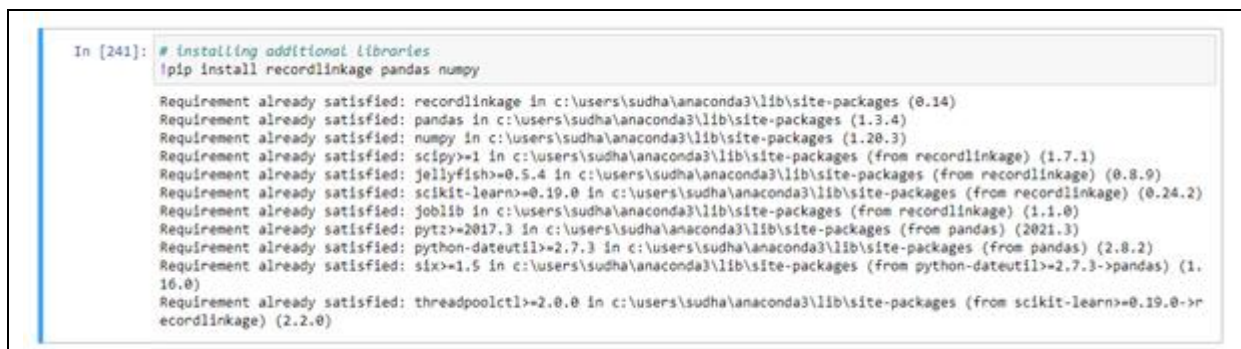
```
In [1]: # installing additional libraries
!pip install recordlinkage pandas numpy

Requirement already satisfied: recordlinkage in c:\users\sudha\anaconda3\lib\site-packages (0.14)
Requirement already satisfied: pandas in c:\users\sudha\anaconda3\lib\site-packages (1.3.4)
Requirement already satisfied: numpy in c:\users\sudha\anaconda3\lib\site-packages (1.20.3)
Requirement already satisfied: joblib in c:\users\sudha\anaconda3\lib\site-packages (from recordlinkage) (1.1.0)
Requirement already satisfied: jellyfish>=0.5.4 in c:\users\sudha\anaconda3\lib\site-packages (from recordlinkage) (0.8.9)
Requirement already satisfied: scipy>=1 in c:\users\sudha\anaconda3\lib\site-packages (from recordlinkage) (1.7.1)
Requirement already satisfied: scikit-learn>=0.19.0 in c:\users\sudha\anaconda3\lib\site-packages (from recordlinkage) (0.24.2)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\sudha\anaconda3\lib\site-packages (from pandas) (2021.3)
Requirement already satisfied: six>=1.5 in c:\users\sudha\anaconda3\lib\site-packages (from python-dateutil>=2.7.3->pandas) (1.16.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\sudha\anaconda3\lib\site-packages (from scikit-learn>=0.19.0->recordlinkage) (2.2.0)

In [2]: # importing required libraries
import pandas as pd
import numpy as np
from recordlinkage import Compare
```

4 Execution of Code for AdaBoost Model

The additional libraries(recordlinkage, pandas, numpy)required to implement the code were installed.



This screenshot shows a single code cell from a Jupyter Notebook. The code cell is labeled 'In [241]:' and contains a comment '# installing additional libraries' followed by the command '!pip install recordlinkage pandas numpy'. The output of this command is displayed below the code, showing that all required packages are already installed: recordlinkage (0.14), pandas (1.3.4), numpy (1.20.3), scipy (1.7.1), jellyfish (0.8.9), scikit-learn (0.24.2), python-dateutil (2021.3), joblib (1.1.0), six (2.8.2), and threadpoolctl (2.2.0).

```
In [241]: # installing additional libraries
!pip install recordlinkage pandas numpy

Requirement already satisfied: recordlinkage in c:\users\sudha\anaconda3\lib\site-packages (0.14)
Requirement already satisfied: pandas in c:\users\sudha\anaconda3\lib\site-packages (1.3.4)
Requirement already satisfied: numpy in c:\users\sudha\anaconda3\lib\site-packages (1.20.3)
Requirement already satisfied: scipy>=1 in c:\users\sudha\anaconda3\lib\site-packages (from recordlinkage) (1.7.1)
Requirement already satisfied: jellyfish>=0.5.4 in c:\users\sudha\anaconda3\lib\site-packages (from recordlinkage) (0.8.9)
Requirement already satisfied: scikit-learn>=0.19.0 in c:\users\sudha\anaconda3\lib\site-packages (from recordlinkage) (0.24.2)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\sudha\anaconda3\lib\site-packages (from pandas) (2021.3)
Requirement already satisfied: joblib in c:\users\sudha\anaconda3\lib\site-packages (from recordlinkage) (1.1.0)
Requirement already satisfied: six>=1.5 in c:\users\sudha\anaconda3\lib\site-packages (from python-dateutil>=2.7.3->pandas) (1.16.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\sudha\anaconda3\lib\site-packages (from scikit-learn>=0.19.0->recordlinkage) (2.2.0)
```

The required libraries were imported and dataset from two data sources (CMDB & UDDR) was loaded.



This screenshot shows two code cells from a Jupyter Notebook. Cell [242] contains a comment '# importing required libraries' followed by a series of import statements: 'import pandas as pd', 'import numpy as np', 'from recordlinkage import Compare', 'from recordlinkage.index import Block', 'from sklearn.model_selection import train_test_split', 'import os', 'import glob', 'import itertools', 'from sklearn.tree import DecisionTreeClassifier', 'from sklearn.metrics import classification_report, confusion_matrix', 'import matplotlib.pyplot as plt', 'from matplotlib import style', and 'import warnings'. It also includes 'warnings.filterwarnings("ignore")' and 'style.use("fivethirtyeight")'. Cell [243] contains a comment '# Loading datasets' followed by two lines of code: 'dataset = pd.read_excel("../dataset/test_cmdb.xlsx")' and 'uddr = pd.read_excel("../dataset/test_UDDR.xlsx")'.

```
In [242]: # importing required libraries
import pandas as pd
import numpy as np
from recordlinkage import Compare
from recordlinkage.index import Block
from sklearn.model_selection import train_test_split
import os
import glob
import itertools
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report, confusion_matrix
import matplotlib.pyplot as plt
from matplotlib import style
import warnings

warnings.filterwarnings("ignore")
style.use("fivethirtyeight")

In [243]: # Loading datasets
dataset = pd.read_excel("../dataset/test_cmdb.xlsx")
uddr = pd.read_excel("../dataset/test_UDDR.xlsx")
```

Then some descriptive analysis was performed on the dataset, and it was checked for null values. The invalid IPs were also removed from the dataset.

```
In [244]: # displaying top-5 rows
dataset.head()
```

```
Out[244]:
```

	FQDN	IP_Address	Asset_State	Device_Subtype	Device_Discovery_Source
0	pc1apsconsole	10.58.90.28	Installed	NaN	ServiceNow
1	ps3apsdev	10.58.218.68	Installed	IP Firewall	Nlyte
2	pc1apsconsole	10.58.90.28	Installed	IP Switch	NaN
3	pc1aed	10.58.90.29	Installed	NaN	NaN
4	ps3aed	10.58.218.67	Installed	NaN	NaN

```
In [245]: uddr.head()
```

```
Out[245]:
```

	FQDN	Host	IP Address
0	crkibypass.pt.abc.com	crkibypass	10.41.184.80
1	dursrsaed.us.abc.com	dursrsaed	10.105.27.32
2	xmnpinwepbro01.xmn.apac.abc.com	xmnpinwepbro01	10.102.221.21
3	usscz01-dc-0204-fw-abcf01.networks.abc.com	usscz01-dc-0204-fw-abcf01	10.202.129.52
4	icc8svcpafw1.ha1	icc8svcpafw1	10.114.130.185\n192.168.1.1

```
In [246]: # shape of the datasets
uddr.shape, dataset.shape
```

```
Out[246]: ((2791, 3), (2693, 5))
```

```
In [247]: # discriptive analysis
dataset.describe()
```

```
Out[247]:
```

	FQDN	IP_Address	Asset_State	Device_Subtype	Device_Discovery_Source
count	2693	2666	2693	701	2336
unique	2619	2502	3	10	15
top	pc1apsconsole	0.0.0.1	Installed	IP Firewall	Solarwinds
freq	2	24	2679	388	997

```
In [248]: # preprocessing dataset
print("Checking for null values")
dataset.isnull().sum()
```

Checking for null values

```
Out[248]: FQDN      0
IP_Address  27
Asset_State  0
Device_Subtype  1992
Device_Discovery_Source  357
dtype: int64
```

```
In [249]: # filling null values
dataset["Device_Discovery_Source"].fillna(method='ffill', inplace = True)
dataset["Device_Subtype"].fillna(method='ffill', inplace = True)
```

```
In [250]: dataset.dropna(subset=["IP_Address"],axis=0, inplace=True)
```

The indexing was done using blocking. Then record pairs were formed and duplicates were dropped.

```
In [251]: dataset.isnull().sum()

Out[251]: FQDN                0
IP_Address                0
Asset_State               0
Device_Subtype            1
Device_Discovery_Source   0
dtype: int64
```

```
In [252]: # checking dataset shape
dataset.shape

Out[252]: (2666, 5)
```

```
In [253]: dataset = dataset.astype(str).apply(lambda x: x.str.upper())
```

```
In [254]: index = Block(on="FQDN")
fqdnIndex = index.index(dataset)
```

```
In [255]: print("Table Records: {} records, No of Pairs: {} pairs".format(dataset.shape[0], len(fqdnIndex)))

Table Records: 2666 records, No of Pairs: 74 pairs
```

```
In [256]: # dropping duplicates
fqdnIndexPairs = fqdnIndex.drop_duplicates(keep="first")
```

```
In [257]: fqdnIndexPairs
```

```
Out[257]: MultiIndex([( 2, 0),
(120, 118),
(371, 210),
(377, 219),
(289, 226),
(271, 227),
(344, 234),
(288, 235),
(346, 243),
(277, 245),
(336, 246),
(262, 249),
(327, 251),
(263, 252),
(372, 260),
(284, 261),
(286, 264),
(385, 268),
(805, 804)])
```

The comparison vector was created, and similarity score is calculated using JaroWinkler method

```
In [19]: compare = Compare()
compare.string('FQDN','FQDN', method='jarowinkler', label = 'FQDN_score')
compare.string('IP_Address','IP_Address', method='jarowinkler', label = 'IP_Address_score')
compare.string('Asset_State','Asset_State', method='jarowinkler', label = 'Asset_State_score')
compare.string('Device_Subtype','Device_Subtype', method='jarowinkler', label = 'Device_Subtype_score')
compare.string('Device_Discovery_Source','Device_Discovery_Source', method='jarowinkler', label = 'Device_Discovery_Source_score')
comparison_vectors = compare.compare(fqdnIndexPairs.dataset)
```

```
In [20]: comparison_vectors.head(5)
```

```
Out[20]:
```

		FQDN_score	IP_Address_score	Asset_State_score	Device_Subtype_score	Device_Discovery_Source_score
2	0	1.0	1.0	1.0	0.0	0.433333
120	118	1.0	1.0	1.0	1.0	0.465079
371	210	1.0	1.0	1.0	1.0	0.465079
377	219	1.0	1.0	1.0	1.0	0.465079
289	226	1.0	1.0	1.0	1.0	0.465079

Dataset Labelling was done based on thershold value.

```
In [22]: # Labeling based on threshold
scores = np.average(
    comparison_vectors.values,
    axis=1,
    weights=[30, 10, 5, 10, 30])
scored_comparison_vectors = comparison_vectors.assign(score=scores)
```

```
In [23]: scored_comparison_vectors.head(5)
```

```
Out[23]:
```

		FQDN_score	IP_Address_score	Asset_State_score	Device_Subtype_score	Device_Discovery_Source_score	score
2	0	1.0	1.0	1.0	0.0	0.433333	0.682353
120	118	1.0	1.0	1.0	1.0	0.465079	0.811204
371	210	1.0	1.0	1.0	1.0	0.465079	0.811204
377	219	1.0	1.0	1.0	1.0	0.465079	0.811204
289	226	1.0	1.0	1.0	1.0	0.465079	0.811204

```
In [24]: scored_comparison_vectors.score.unique()
```

```
Out[24]: array([0.68235294, 0.81120448, 0.85882353, 0.75030133, 0.79298701,
0.8       , 0.75297513, 1.       , 0.75160286, 0.75676089,
0.75837931, 0.80504202, 0.76454178, 0.71657471, 0.80375746,
0.77525677, 0.7262598 , 0.75807869, 0.7163625 , 0.85025606,
0.74104207, 0.73355733, 0.7171413 , 0.84468565, 0.83741411,
0.79607843, 0.79572193, 0.79215686])
```

If the score was greater than or equal to 0.85, it was considered a match.

```
In [24]: scored_comparison_vectors.score.unique()
```

```
Out[24]: array([0.68235294, 0.81120448, 0.85882353, 0.75030133, 0.79298701,
0.8       , 0.75297513, 1.       , 0.75160286, 0.75676089,
0.75837931, 0.80504202, 0.76454178, 0.71657471, 0.80375746,
0.77525677, 0.7262598 , 0.75807869, 0.7163625 , 0.85025606,
0.74104207, 0.73355733, 0.7171413 , 0.84468565, 0.83741411,
0.79607843, 0.79572193, 0.79215686])
```

```
In [25]: matches = comparison_vectors[scored_comparison_vectors['score'] >= 0.85]
matches.head(5)
```

```
Out[25]:
```

		FQDN_score	IP_Address_score	Asset_State_score	Device_Subtype_score	Device_Discovery_Source_score	score
271	227	1.0	1.0	1.0	1.000000	0.600000	
1128	1074	1.0	1.0	1.0	1.000000	1.000000	
1106	1091	1.0	1.0	1.0	1.000000	0.600000	
1554	1477	1.0	1.0	1.0	1.000000	0.600000	
1800	1785	1.0	1.0	1.0	0.603367	0.707937	

Then the data was compared with SW dataset which is the source of truth and contains only unique records. Only the records present in SW were stored. The duplicate records were rejected.


```
In [26]: # creating labels
truth = pd.read_excel("../dataset/test_SW.xlsx")
```

```
In [27]: trueIndexPairs = Block(on="FQDN").index(truth)
```

```
In [28]: trueIndexPairs
```

```
Out[28]: MultiIndex([( 616,  615),
                    (2031, 2030),
                    (2080, 2079),
                    (4149, 4148),
                    (5489, 5488),
                    (5491, 5490),
                    (5493, 5492),
                    (5495, 5494),
                    (5497, 5496),
                    (5803, 5802),
                    (5805, 5804),
                    (5807, 5806),
                    (5809, 5808),
                    (5811, 5810)],
                    )
```

Then centroid function was defined to differentiate the dataset between duplicates and non-duplicates.

```
In [30]: centroids = {}
K = 2
```

```
In [31]: for i in range(K):
          centroids[i] = duplicate_pairs.iloc[i,:].values
```

Input and output variables as well as test and train data variables and length were defined. AdaBoost class was defined.

```
In [33]: # input and output
x = []
y = []
for label, values in predicted_labels.items():
    for val in values:
        x.append(val)
        y.append(label)
```

```
In [34]: x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.2,random_state=87752)
```

```
In [35]: len(x_train), len(x_test)
```

```
Out[35]: (59, 15)
```

```
In [36]: class Boosting:
          def __init__(self, feature, labels, T, test_feature, test_labels):
              self.T = T
              self.feature = feature
              self.labels = labels
              self.test_feature = test_feature
              self.test_labels = test_labels

              self.alphas = list()
              self.models = list()
```

After executing, the accuracy of 87% was obtained.

```
In [38]: C = 0
for idx in range(len(y_test)):
    if y_test[idx] == list(model.predictions)[idx]:
        C += 1

print("Accuracy rate: %0.3f" %(C / float(len(y_test)) * 100.0) + u"\u0025")

Accuracy rate: 86.667%
```

Also, Precision, Recall and F1 score were calculated.

```
print("Precision is ", precision)
print("Recall is ", recall)
print("F1 Score is ", f1_score)

[1 1 1 1 1 1 1 1 1 1 1 1 0 1 1] [0 1 1 1 1 1 1 1 1 1 1 1 0 1 0]
tp = 12
tn = 1
fn = 2
fp = 0
Precision is 100.0
Recall is 85.71428571428571
F1 Score is 92.30769230769229
```

5 Internship Activity Report

Student Name: Sudha Gaurinath Koride.

Student number: 20196083

Company: Dell Technologies, Ireland.

Month Commencing: September,2021.

- Initial meeting with Mentors and Senior Manager of the Infrastructure Management & Compliance team and briefing of the project.
- Attended IT Induction and raised requests to get access to the dashboards and tools.
- Attended HR Induction and got acquainted with all the necessary tools required at the workplace.
- Introduction to the tools/dashboards: SolarWinds, CMDB, Kenna, Asset Registry.
- Started working on Asset Inventory Audit to make sure that all the devices are uniform across each tool.
- Completed Splunk Fundamentals & Splunk Infrastructure training.

Student Name: Sudha Gaurinath Koride.

Student number: 20196083

Company: Dell Technologies, Ireland.

Month Commencing: October,2021.

- Completed training on Data Protection, Phishing, Ransomware, Incident Reporting as part of Security Awareness Month program.
- Completed “Protecting Against Ransomware” as part of Dell Security Awareness & Training Program.
- Completed Policies, Standards and Best Practices for a Secure Workplace course from the Dell Global Ethics & Compliance Training Program.
- Completed Dell Technologies’ Code of Conduct Course.
- Completed “Be the Change Essentials” training.

Student Name: Sudha Gaurinath Koride.

Student number: 20196083

Company: Dell Technologies, Ireland.

Month Commencing: November,2021.

- Attended Palo Alto Ignite21 event and attended the following information sessions in it:
 - 1.Firewall Product Training by keynote speakers Baba Diao, Lars Meyer & Robert Donohoe. PAN-OS Controls Applications with App-ID, Controlling Access to network resources using User-ID & Troubleshooting Firewalls with Flow Logic.
 - 2.Building a Cloud Security Program Based on the NIST CSF by Ankur Shah & Birat Niraula.
 - 3.Top 5 Real-Life NetOps Automation Use Cases by Lior Kolnik & Rushton James.

4. How do I Protect my Attack Surface? By Madhuresh Anur.
 5. Anatomy of a Cyber Attack by Danny Milrad.
 6. Palo Alto Networks Certified Network Security Administrator (PCNSA) session by Michael Kalish.
- Attended training session on Kenna.

Student Name: Sudha Gaurinath Koride.

Student number: 20196083

Company: Dell Technologies, Ireland.

Month Commencing: December, 2021.

- Understood the architecture of network devices and the flow of the data through various network devices and how they communicate with each other.
- Extracted the data from SolarWinds and CMDB dashboards.
- Extracted data from UDDR with the help of mentors.
- Encrypted the data extracted using character substitution and shuffling.
- Studied the data extracted and investigated the reasons for duplication of data in order to proceed with the Research Project.
- Updated slides with the details of assets (and reported issues if found any) from SolarWinds, CMDB, Kenna, Asset Registry every week to ensure the uniformity of assets over each tool.
- Understood how to perform sprints over JIRA using agile methodology and assisted mentor with the same.

Employer Comments:

Sudha slotted into the team very well during her internship

Student Signature:



Date: 16th Dec, 2021.

Industry Supervisor Signature: Catherine Minogue

Date: 20/12/21

6 Internship Feedback

Internal Use - Confidential

From: Minogue, Catherine <Catherine.Monogue@Dell.com>

Sent: 20 December 2021 11:24

To: Koride, Sudha

Subject: RE: Internship Activity Report

Hi Sudha,

Please see attached updated report with my signature and comment

Many thanks for all your work during your internship

Regards

Catherine

Internal Use - Confidential

References

- [1] <https://docs.anaconda.com/anaconda/navigator/index.html>. Accessed: 4th Jan2022.
- [2] <https://jupyter-notebook.readthedocs.io/en/stable/> Accessed: 4th Jan2022.

-----END OF REPORT-----