Configuration Manual

MSc Internship
MSc Cyber Security

Nelson Seyi Ayo-Akere
X18172521

School of Computing
National College of Ireland

Supervisor: Christos Grecos

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: **NELSON SEYI AYO-AKERE** ……………………………………………………………………………………………

Student ID: **X18172521** ……………………………………………………………………………………………

Program: **MSC CYBER SECURITY** ………………………………………………………… **Year:** **2020** …………………………..

Module: **ACADEMIC INTERNSHIP** ……………………………………………………………………………………………

Lecturer: **CHRISTOS GRECOS** ……………………………………………………………………………………………

Submission Due Date: **12^{TH} December 2019** ……………………………………………………………………………………………

Project Title: **Towards an Effective Social Engineering susceptibility detection Model Using Machine Learning on the Online Social Network** ……………………………………………………………………………………………

Word Count: 1643 ……………………………………… **Page Count:** 10 ……………………………………….

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: ……………………………………………………………………………………………

Date: ……………………………………………………………………………………………

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

Nelson Seyi Ayo-Akere
X18172521

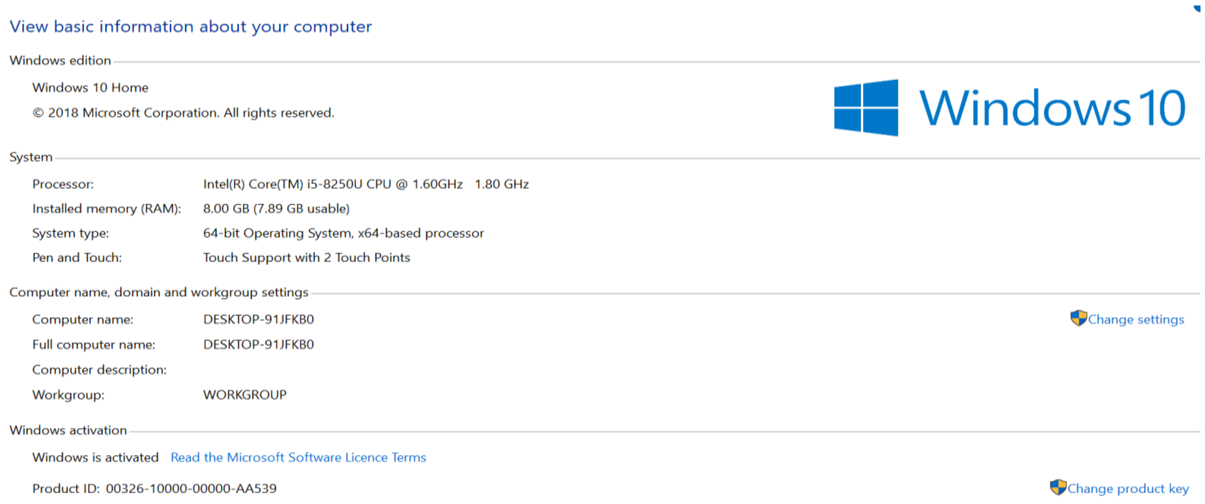
SECTION 1

1.0 Introduction

This manual is to complement the research paper submitted to the national college of Ireland as part of the MSc. In Cyber Security 'Towards an Effective Social Engineering susceptibility detection Model Using Machine Learning on the Online Social Network'. This manual discusses the hardware and software technologies utilized, their application, and a detailed work through the key areas and tasks involved in the development of our social engineering machine learning prediction model (SE-MLPM), so that the project can be replicated any time.

1.1 Hardware Specification

The hardware specification used in this project was carefully selected to handle the task and its requirements. The figure below shows the hardware specification of the computer system used for the installation of necessary software requirements and packages and in the development of the project model social engineering machine learning prediction model SE-MLPM.



View basic information about your computer

Windows edition

Windows 10 Home
© 2018 Microsoft Corporation. All rights reserved.

System

Processor:	Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz
Installed memory (RAM):	8.00 GB (7.89 GB usable)
System type:	64-bit Operating System, x64-based processor
Pen and Touch:	Touch Support with 2 Touch Points

Computer name, domain and workgroup settings

Computer name:	DESKTOP-91JFKB0
Full computer name:	DESKTOP-91JFKB0
Computer description:	
Workgroup:	WORKGROUP

Windows activation

Windows is activated Read the Microsoft Software Licence Terms

Product ID: 00326-10000-00000-AA539

Figure 1 Hardware specification

Detailed software requirements

The table below shows a detailed list of software and packages requirements that will be installed or used during this work through. It is of note that some of this software and packages come by default on installation of Anaconda and Python software, while others are available on install of pandas, scikit-learn joblib and flask. Any of the software and packages can be installed if not in system already by typing the below command on the command line prompt.

Pip install (command)

Anaconda==2019.10 **Flask==1.1.1** **parso==0.5.1** **pathtools==0.1.2** **jupyter-core==4.6.1** **jupyterlab==1.2.3** **jupyterlab-server==1.0.6** **matplotlib** **eli5**	**importlib-metadata==1.1.0** **ipykernel==5.1.3** **ipython==7.10.1** **joblib==0.14.0** **json5==0.8.5** **jupyter-client==5.3.4** **six==1.13.0** **spacy==2.2.3**	**preshed==3.0.2** **prometheus-client==0.7.1** **Python==3.7.5** **sublime text==3.0** **scikit-learn==0.21.3** **scipy==1.3.3** **Send2Trash==1.5.0**	**toolz==0.10.0** **webencodings==0.5.1** **Werkzeug==0.16.0** **zipp==0.6.0** **numpy==1.17.4** **pandas==0.25.3** **pandocfilters==1.4.2** **toml==0.10.0** **notebook==6.0.2**

Figure 2 software specification

SECTION 2

Creating a Folder Environment

The first thing we shall do is to create a folder environment in our system where we can save models and files in, and where we can automatically run files from. The procedure is as follows:

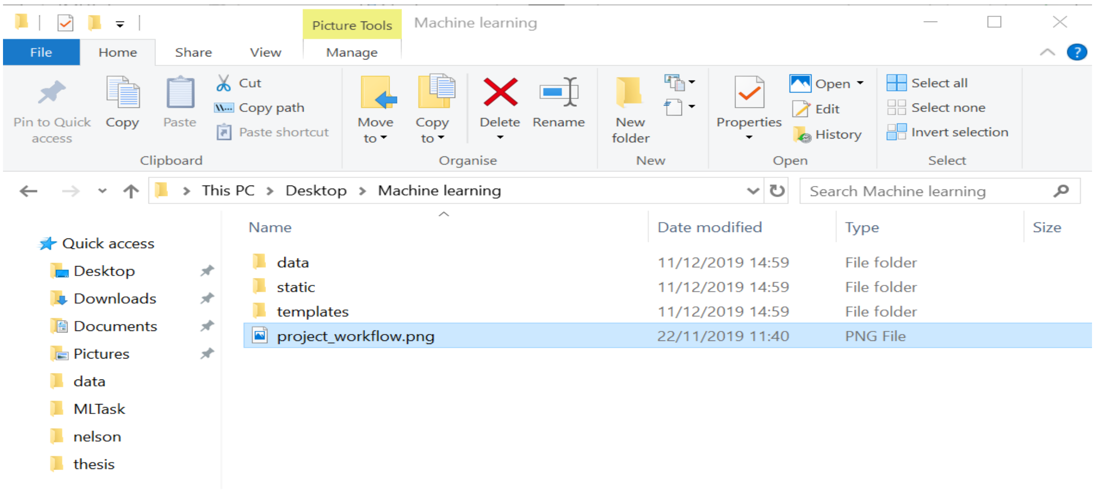
a. Right click on desktop

b. Create new folder and name it e.g. machine

c. Open the folder and create 4 folders namely: **Data**, **Template**, and **Static**

d. Open the static folder and create another folder called **Models** where we will be saving our machine learning models latter on.

e. Also, the schematic of project workflow can be added to the folder.



Name	Date modified	Type	Size
data	11/12/2019 14:59	File folder	
static	11/12/2019 14:59	File folder	
templates	11/12/2019 14:59	File folder	
project_workflow.png	22/11/2019 11:40	PNG File	

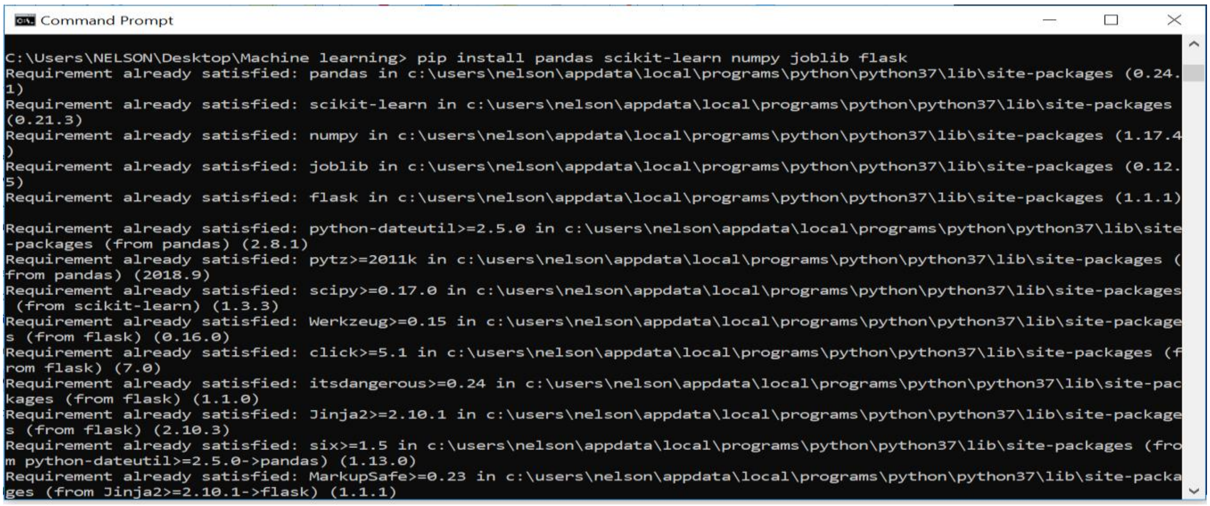
Figure 3 Creating the folder environment

Installations

To install the packages and software simply follow this step

a. go to windows command prompt

b. in the command line type in pip install all the software necessary as shown in the screenshot below.



```
C:\Users\NELSON\Desktop\Machine learning> pip install pandas scikit-learn numpy joblib flask
Requirement already satisfied: pandas in c:\users\nelson\appdata\local\programs\python\python37\lib\site-packages (0.24.
1)
Requirement already satisfied: scikit-learn in c:\users\nelson\appdata\local\programs\python\python37\lib\site-packages
(0.21.3)
Requirement already satisfied: numpy in c:\users\nelson\appdata\local\programs\python\python37\lib\site-packages (1.17.4
)
Requirement already satisfied: joblib in c:\users\nelson\appdata\local\programs\python\python37\lib\site-packages (0.12.
5)
Requirement already satisfied: flask in c:\users\nelson\appdata\local\programs\python\python37\lib\site-packages (1.1.1)

Requirement already satisfied: python-dateutil>=2.5.0 in c:\users\nelson\appdata\local\programs\python\python37\lib\site
-packages (from pandas) (2.8.1)
Requirement already satisfied: pytz>=2011k in c:\users\nelson\appdata\local\programs\python\python37\lib\site-packages (
from pandas) (2018.9)
Requirement already satisfied: scipy>=0.17.0 in c:\users\nelson\appdata\local\programs\python\python37\lib\site-packages
 (from scikit-learn) (1.3.3)
Requirement already satisfied: Werkzeug>=0.15 in c:\users\nelson\appdata\local\programs\python\python37\lib\site-package
s (from flask) (0.16.0)
Requirement already satisfied: click>=5.1 in c:\users\nelson\appdata\local\programs\python\python37\lib\site-packages (f
rom flask) (7.0)
Requirement already satisfied: itsdangerous>=0.24 in c:\users\nelson\appdata\local\programs\python\python37\lib\site-pac
kages (from flask) (1.1.0)
Requirement already satisfied: Jinja2>=2.10.1 in c:\users\nelson\appdata\local\programs\python\python37\lib\site-package
s (from flask) (2.10.3)
Requirement already satisfied: six>=1.5 in c:\users\nelson\appdata\local\programs\python\python37\lib\site-packages (fro
m python-dateutil>=2.5.0->pandas) (1.13.0)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\nelson\appdata\local\programs\python\python37\lib\site-packa
ges (from Jinja2>=2.10.1->flask) (1.1.1)
```

Figure 4 Software and command line installation

SECTION 2
Walkthrough

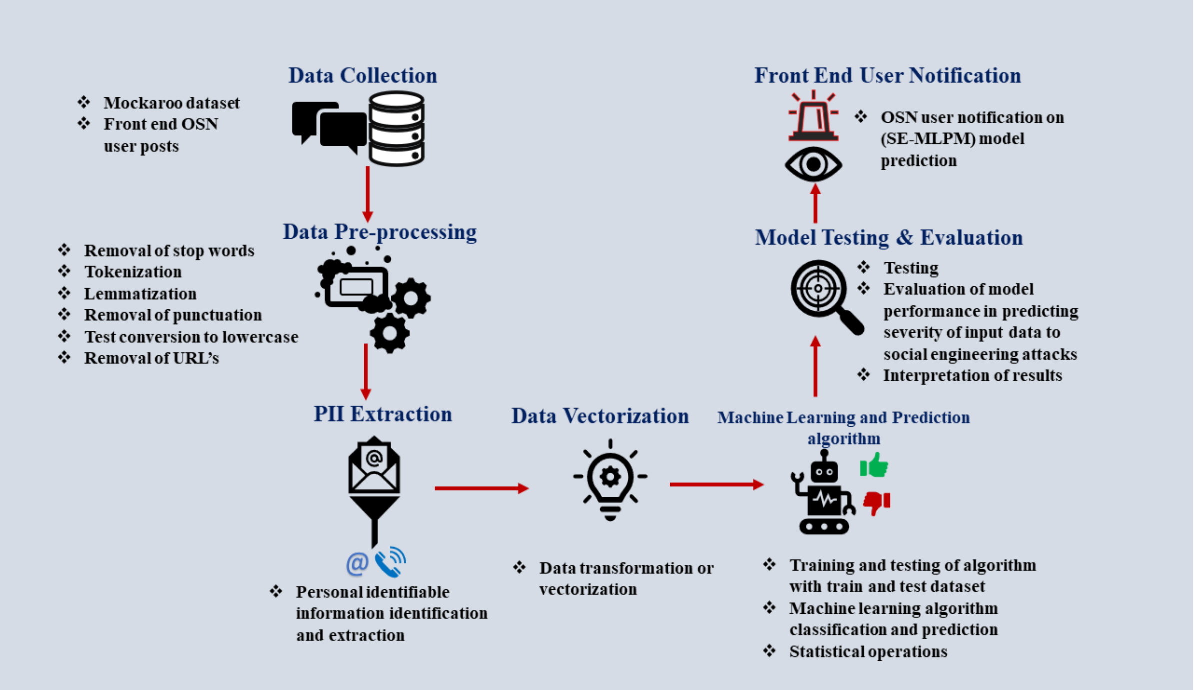

Data Collection
❖ Mockaroo dataset
❖ Front end OSN user posts
Data Pre-processing
❖ Removal of stop words
❖ Tokenization
❖ Lemmatization
❖ Removal of punctuation
❖ Test conversion to lowercase
❖ Removal of URL's
PII Extraction
❖ Personal identifiable information identification and extraction
Data Vectorization
❖ Data transformation or vectorization
Machine Learning and Prediction algorithm
❖ Training and testing of algorithm with train and test dataset
❖ Machine learning algorithm classification and prediction
❖ Statistical operations
Model Testing & Evaluation
❖ Testing
❖ Evaluation of model performance in predicting severity of input data to social engineering attacks
❖ Interpretation of results
Front End User Notification
❖ OSN user notification on (SE-MLPM) model prediction

Figure 5 Workflow of model methodology and implementation

Stage 1: Data collection

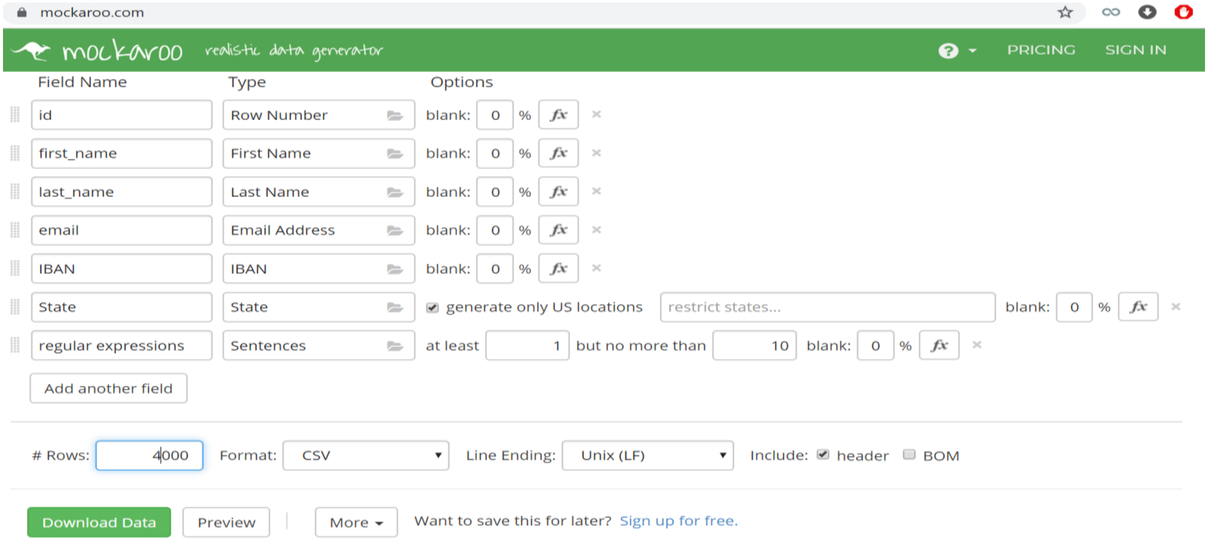
Synthetic data generated was gotten from mockaroo data generator. To navigate and download data from the platform the following process was carried out:

a. In the URL type in www.mockaroo.com [1]

b. Select the necessary field of wanted data from the fields presented, more fields can be added by simply clicking on more fields

the figure below show the web page for mockaroo data generator

c. A total of 4000 rows of data was downloaded in csv format and transported to MS Excel for visualization.



mockaroo.com

Field Name	Type	Options
id	Row Number	blank: 0 %
first_name	First Name	blank: 0 %
last_name	Last Name	blank: 0 %
email	Email Address	blank: 0 %
IBAN	IBAN	blank: 0 %
State	State	☑ generate only US locations restrict states... blank: 0 %
regular expressions	Sentences	at least 1 but no more than 10 blank: 0 %

Add another field

Rows: 4000 Format: CSV Line Ending: Unix (LF) Include: ☑ header □ BOM

Mockaroo data generator

d. The data set is thus saved in the data folder in in the initially created folder (machine).

Stage 2: Next, download the Anaconda software into the system from www.anaconda.com (2019.10 version exe file) and install following the manufacturer's installation procedure. After following the basic manufacturer steps of download, the application is launched from the system by

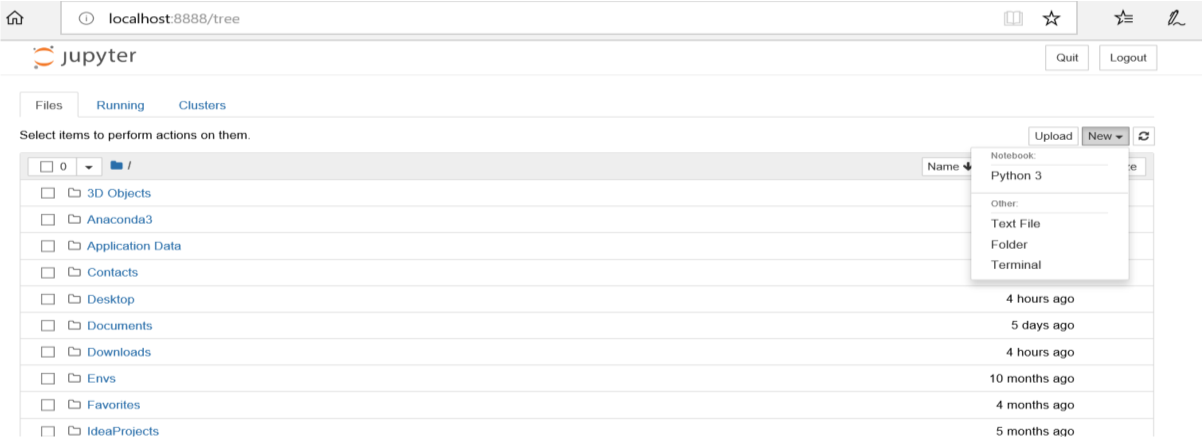
a. Navigating from the search button on the system and search for anaconda

b. Launch both the anaconda prompt panel and navigator

c. Click on the launch button Jupyter notebook 6.0.1 to navigate you to the jupyter notebook environment.

d. There will be an automatic re-direction to the default web browser on the computer system.

e. Click on the new button on the top right corner of the web page and select 'python 3'



localhost:8888/tree

Select items to perform actions on them.

Name	Last Modified
3D Objects	
Anaconda3	
Application Data	
Contacts	
Desktop	4 hours ago
Documents	5 days ago
Downloads	4 hours ago
Envs	10 months ago
Favorites	4 months ago
IdeaProjects	5 months ago

Figure 6 Jupyter notebook environment

f. This will automatically take you to another web page where you can write your python code

g. Rename and save the Jupyter notebook page for easy identification

h. Click on the far-right corner of the page and change the untrusted box to 'trusted'

i. You are now ready to start building your model

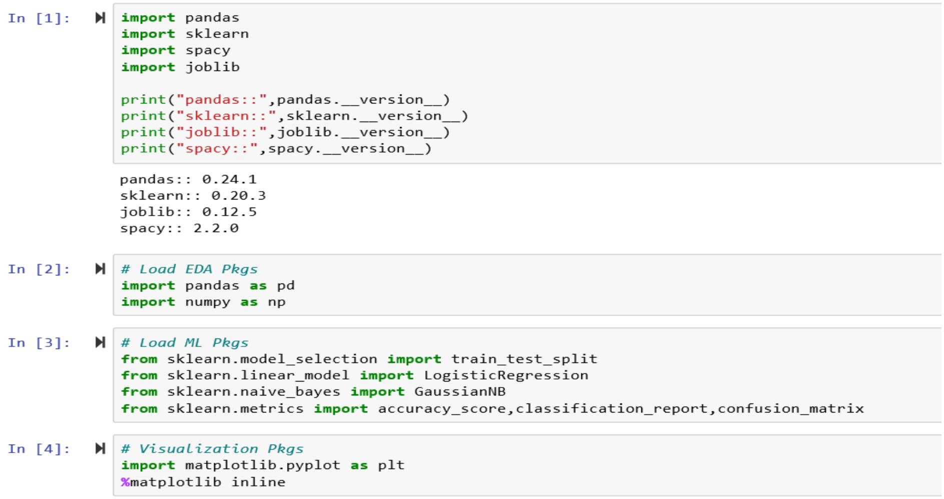
Stage 3: Building the model (SE-MLPM)

This project is conducted to design a social engineering machine learning prediction model (SE-MLPM) for the detection and extraction of PII in OSN user posts utilizing natural language processing (bag-of-words) and thereafter vectorize dataset into vectors making use of the term frequency inverse document frequency (TF-IDF) vector space modelling technique and then classify, label and predict levels of post susceptibility to social engineering attacks in addition to revealing the PII discovered to the OSN user and recommending to the user if the post should go live or not, based on PII count recovered from the post ranging from a high susceptibility level to a no susceptibility level using the logistic regression classification algorithm [2].

Packages importation

a. Import all the necessary packages for development of the model.

b. Load the exploratory data packages, machine learning packages and visualization packages, as shown in the figure below.



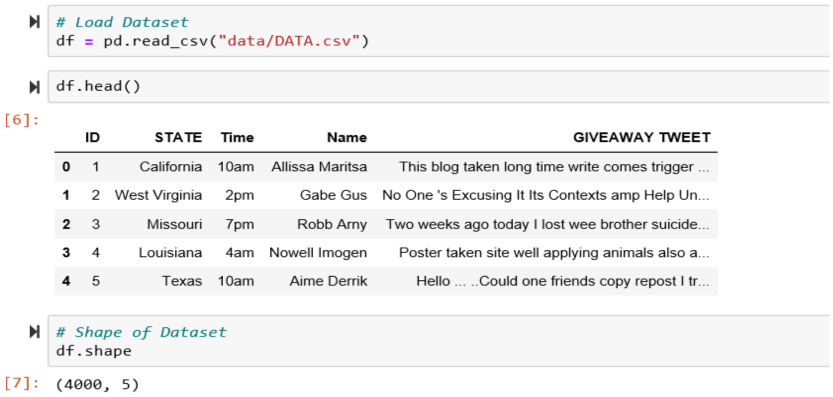
```
In [1]:  import pandas
         import sklearn
         import spacy
         import joblib

         print("pandas::",pandas.__version__)
         print("sklearn::",sklearn.__version__)
         print("joblib::",joblib.__version__)
         print("spacy::",spacy.__version__)

         pandas:: 0.24.1
         sklearn:: 0.20.3
         joblib:: 0.12.5
         spacy:: 2.2.0

In [2]:  # Load EDA Pkgs
         import pandas as pd
         import numpy as np

In [3]:  # Load ML Pkgs
         from sklearn.model_selection import train_test_split
         from sklearn.linear_model import LogisticRegression
         from sklearn.naive_bayes import GaussianNB
         from sklearn.metrics import accuracy_score,classification_report,confusion_matrix

In [4]:  # Visualization Pkgs
         import matplotlib.pyplot as plt
         %matplotlib inline
```

Figure 7 package importation

Loading the various packages

a. The next step is to load the data saved in our initially created data. RUN df = pd.read_csv("data/DATA.csv").

b. Check the shape of the dataset. RUN df.shape to evaluate how many columns and rows are available.



```
# Load Dataset
df = pd.read_csv("data/DATA.csv")
```

```
df.head()
```

[6]:

	ID	STATE	Time	Name	GIVEAWAY TWEET
0	1	California	10am	Allissa Maritsa	This blog taken long time write comes trigger ...
1	2	West Virginia	2pm	Gabe Gus	No One 's Excusing It Its Contexts amp Help Un...
2	3	Missouri	7pm	Robb Arny	Two weeks ago today I lost wee brother suicide...
3	4	Louisiana	4am	Nowell Imogen	Poster taken site well applying animals also a...
4	5	Texas	10am	Aime Derrik	Hello ... ..Could one friends copy repost I tr...

```
# Shape of Dataset
df.shape
```

```
[7]: (4000, 5)
```

Figure 8 loading dataset

Data pre-processing

The next step is to pre-process our data using spacy

a. Import Spacey to extract entities

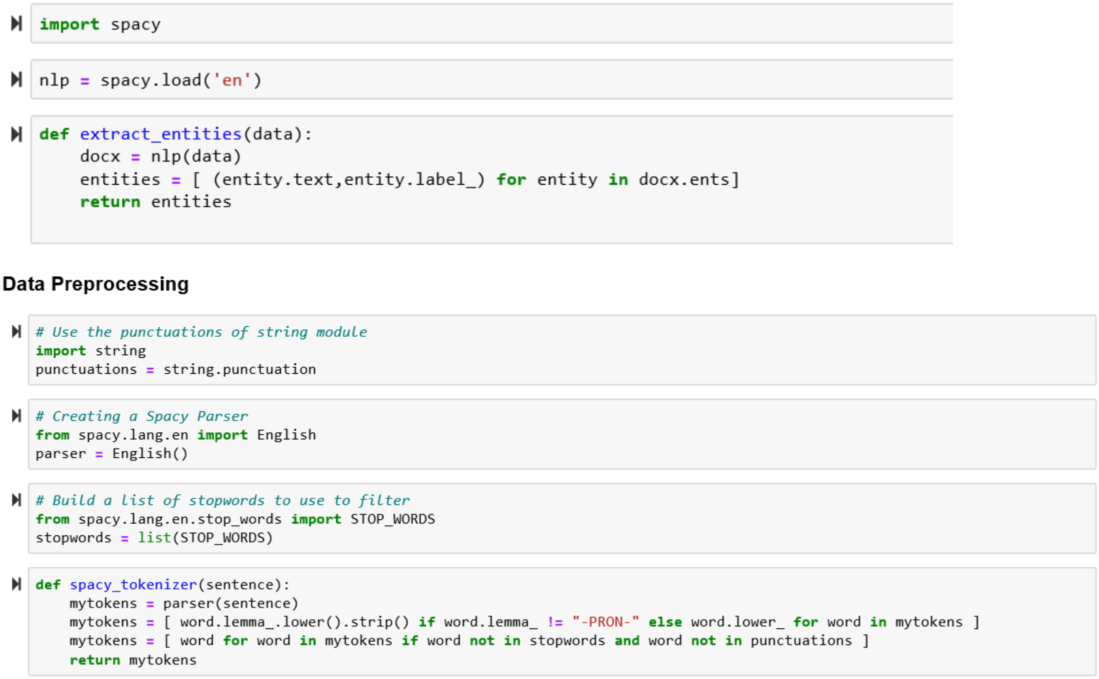
b. we import string

c. Then we create a spacy parser

d. Build a list of stopwords to use to filter

e. Define spacy tokenizer and lemmatization

The individual programming commands are as shown in the figure below



```
import spacy
```

```
nlp = spacy.load('en')
```

```
def extract_entities(data):
    docx = nlp(data)
    entities = [ (entity.text,entity.label_) for entity in docx.ents]
    return entities
```

Data Preprocessing

```
# Use the punctuations of string module
import string
punctuations = string.punctuation
```

```
# Creating a Spacy Parser
from spacy.lang.en import English
parser = English()
```

```
# Build a list of stopwords to use to filter
from spacy.lang.en.stop_words import STOP_WORDS
stopwords = list(STOP_WORDS)
```

```
def spacy_tokenizer(sentence):
    mytokens = parser(sentence)
    mytokens = [ word.lemma_.lower().strip() if word.lemma_ != "-PRON-" else word.lower_ for word in mytokens ]
    mytokens = [ word for word in mytokens if word not in stopwords and word not in punctuations ]
    return mytokens
```

Figure 9 Data pre-processing

Identification and extraction of PII

Since we are more interested in some PII, we will be taking the give away tweets and extract PII from it.

a. RUN df.columns
c. RUN df.columns = df.columns.str.lower().str.replace(' ','_')
d. RUN df.columns AGAIN
e. RUN f.rename(columns={"giveaway_tweet":"giveaway_tweets"},inplace=True)

Build an email, phone number and IBAN REGEX FUNCTION

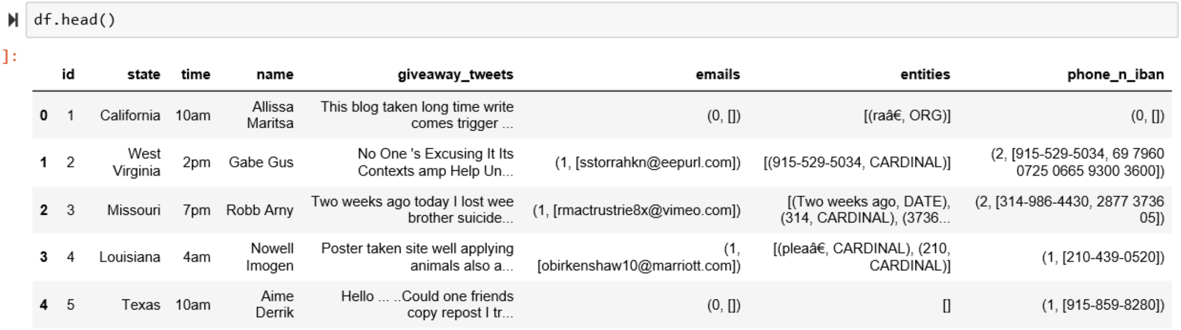
f. Run import re
g. Define the email, phone number and IBAN regex
h. def extract_email(data):
results = email_regex.findall(data)
num_of_results = len(results)
return num_of_results,results

I. def extract_phone_num(data):
   results = phone_num_regex.findall(data)
   results2 = phone_num_regex2.findall(data)
   num_of_results = len(results)
   return num_of_results,results,results2

J. def extract_custom_num(data):
   results = phone_num_regex_n_iban.findall(data)
   num_of_results = len(results)
   return num_of_results,results

Get PII count

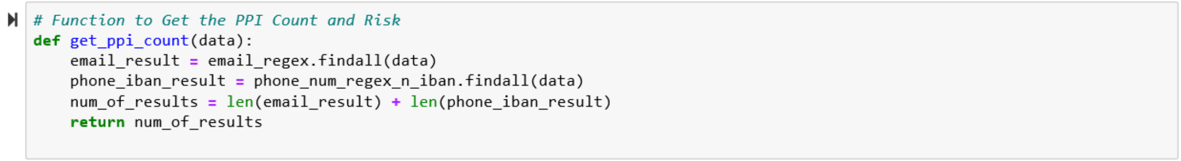
a. Run df.head()



```
df.head()
```

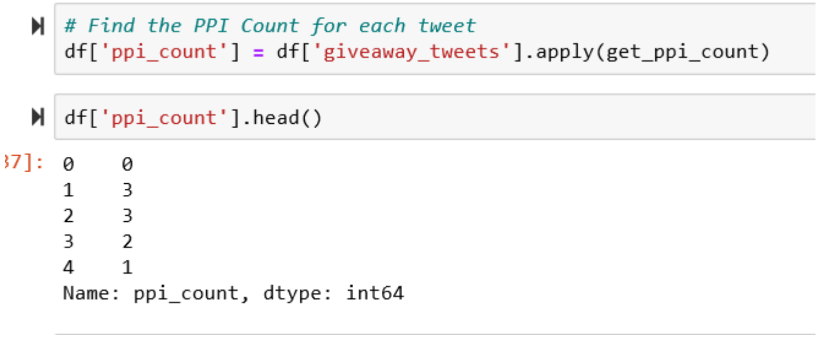
]:

	id	state	time	name	giveaway_tweets	emails	entities	phone_n_iban
0	1	California	10am	Allissa Maritsa	This blog taken long time write comes trigger ...	(0, [])	[(raâ€, ORG)]	(0, [])
1	2	West Virginia	2pm	Gabe Gus	No One 's Excusing It Its Contexts amp Help Un...	(1, [sstorrahkn@eepurl.com])	[(915-529-5034, CARDINAL)]	(2, [915-529-5034, 69 7960 0725 0665 9300 3600])
2	3	Missouri	7pm	Robb Arny	Two weeks ago today I lost wee brother suicide...	(1, [rmactrustrie8x@vimeo.com])	[(Two weeks ago, DATE), (314, CARDINAL), (3736...	(2, [314-986-4430, 2877 3736 05])
3	4	Louisiana	4am	Nowell Imogen	Poster taken site well applying animals also a...	(1, [obirkenshaw10@marriott.com])	[(pleaâ€, CARDINAL), (210, CARDINAL)]	(1, [210-439-0520])
4	5	Texas	10am	Aime Derrik	Hello ... ..Could one friends copy repost I tr...	(0, [])	[]	(1, [915-859-8280])



```
# Function to Get the PPI Count and Risk
def get_ppi_count(data):
    email_result = email_regex.findall(data)
    phone_iban_result = phone_num_regex_n_iban.findall(data)
    num_of_results = len(email_result) + len(phone_iban_result)
    return num_of_results
```

b. Run f['ppi_count'] = df['giveaway_tweets'].apply(get_ppi_count)df['ppi_count'].head()

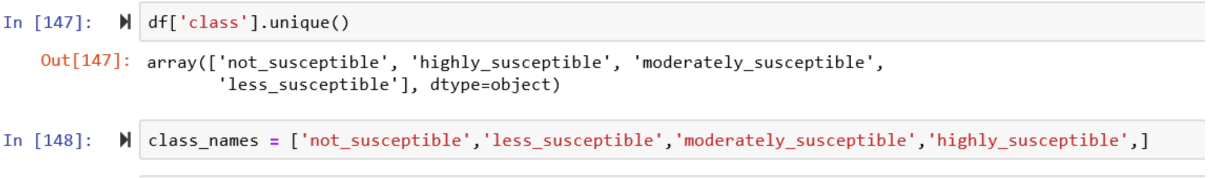


```
# Find the PPI Count for each tweet
df['ppi_count'] = df['giveaway_tweets'].apply(get_ppi_count)
```

```
df['ppi_count'].head()
```

```
37]: 0    0
     1    3
     2    3
     3    2
     4    1
     Name: ppi_count, dtype: int64
```

c. labeling the count we run the command as shown on the figure below

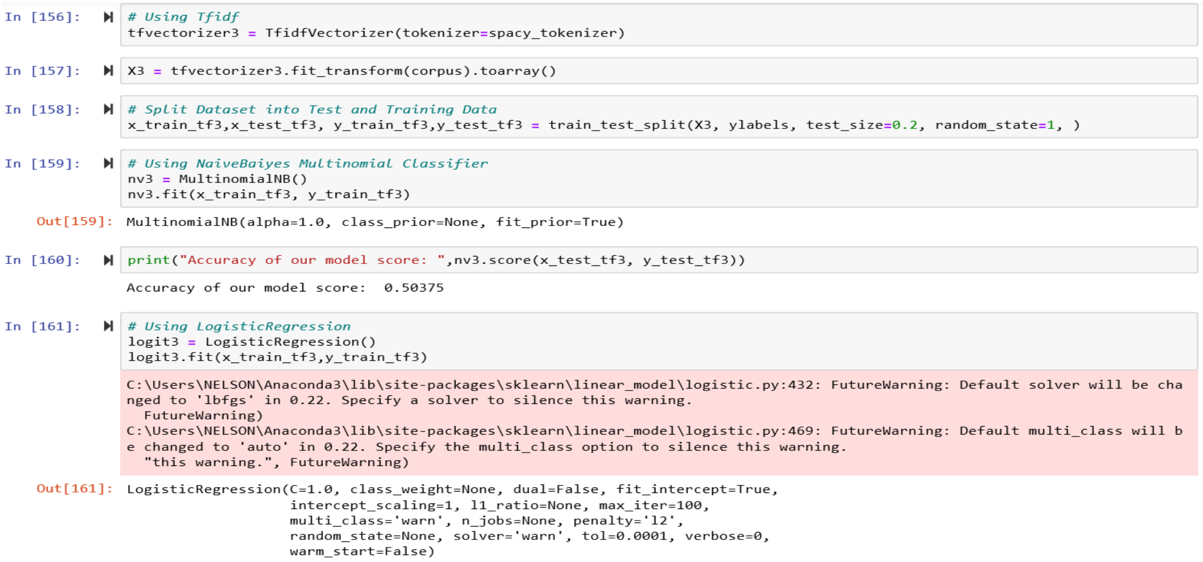


```
In [147]:  df['class'].unique()

Out[147]:  array(['not_susceptible', 'highly_susceptible', 'moderately_susceptible',
                  'less_susceptible'], dtype=object)

In [148]:  class_names = ['not_susceptible','less_susceptible','moderately_susceptible','highly_susceptible',]
```

d. Vectorize the data with spacy tokenizer to test and train data as shown in the figure below

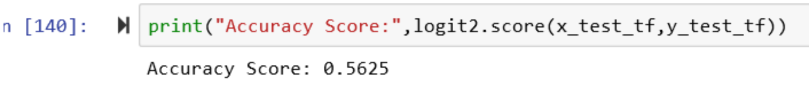
e. Apply logistic regression classifier

f. Tune the model to the best possible output



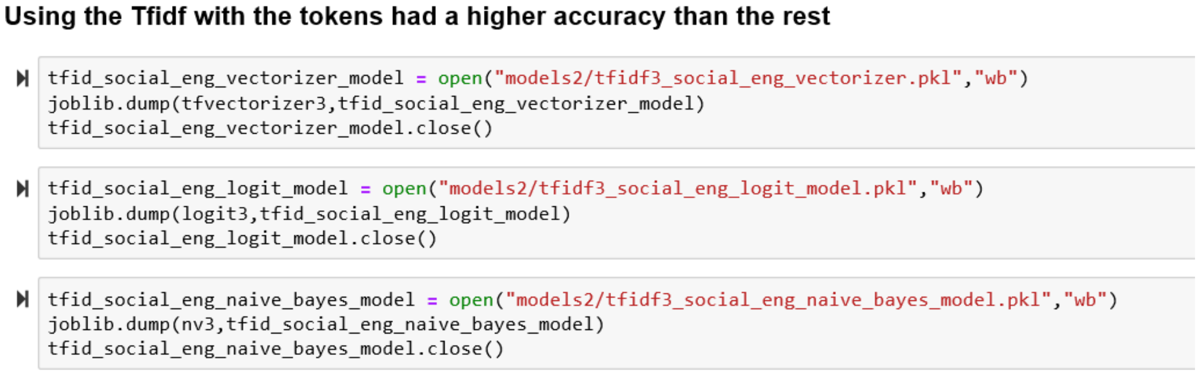
```
In [156]:  # Using Tfidf
           tfvectorizer3 = TfidfVectorizer(tokenizer=spacy_tokenizer)

In [157]:  X3 = tfvectorizer3.fit_transform(corpus).toarray()

In [158]:  # Split Dataset into Test and Training Data
           x_train_tf3,x_test_tf3, y_train_tf3,y_test_tf3 = train_test_split(X3, ylabels, test_size=0.2, random_state=1, )

In [159]:  # Using NaiveBaiyes Multinomial Classifier
           nv3 = MultinomialNB()
           nv3.fit(x_train_tf3, y_train_tf3)

  Out[159]: MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)

In [160]:  print("Accuracy of our model score: ",nv3.score(x_test_tf3, y_test_tf3))

           Accuracy of our model score:  0.50375

In [161]:  # Using LogisticRegression
           logit3 = LogisticRegression()
           logit3.fit(x_train_tf3,y_train_tf3)

           C:\Users\NELSON\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning: Default solver will be cha
           nged to 'lbfgs' in 0.22. Specify a solver to silence this warning.
             FutureWarning)
           C:\Users\NELSON\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:469: FutureWarning: Default multi_class will b
           e changed to 'auto' in 0.22. Specify the multi_class option to silence this warning.
             "this warning.", FutureWarning)

  Out[161]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                               intercept_scaling=1, l1_ratio=None, max_iter=100,
                               multi_class='warn', n_jobs=None, penalty='l2',
                               random_state=None, solver='warn', tol=0.0001, verbose=0,
                               warm_start=False)
```

g. Print result of SE-MLPM MODEL



```
n [140]:  print("Accuracy Score:",logit2.score(x_test_tf,y_test_tf))

          Accuracy Score: 0.5625
```

h. Save the model using run joblib



Using the Tfidf with the tokens had a higher accuracy than the rest

```
tfid_social_eng_vectorizer_model = open("models2/tfidf3_social_eng_vectorizer.pkl","wb")
joblib.dump(tfvectorizer3,tfid_social_eng_vectorizer_model)
tfid_social_eng_vectorizer_model.close()
```

```
tfid_social_eng_logit_model = open("models2/tfidf3_social_eng_logit_model.pkl","wb")
joblib.dump(logit3,tfid_social_eng_logit_model)
tfid_social_eng_logit_model.close()
```

```
tfid_social_eng_naive_bayes_model = open("models2/tfidf3_social_eng_naive_bayes_model.pkl","wb")
joblib.dump(nv3,tfid_social_eng_naive_bayes_model)
tfid_social_eng_naive_bayes_model.close()
```

I. Download the worksheet

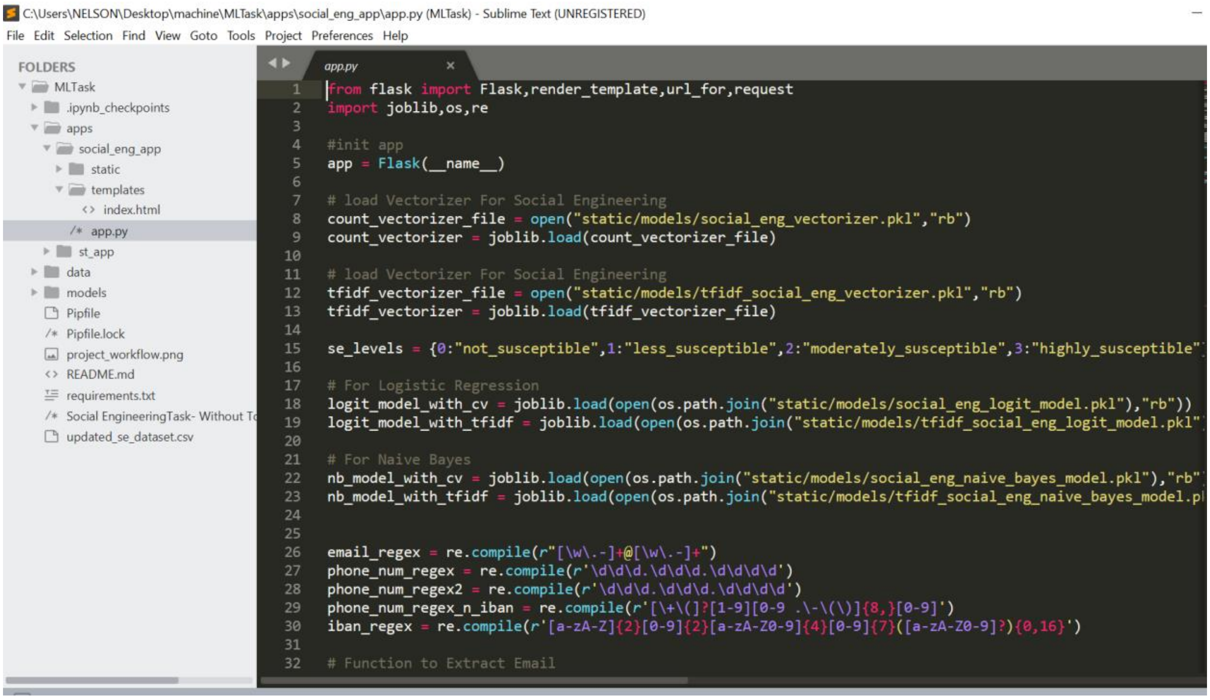
click on file>download as >ipynb >save as > models folder

J. Integrate model with flask

>Launch sublime text icon
>Import folder into sublime
>Click on file > new file > save as app.py
>Import flask on app.py
>Create another file and name it index.html

Render index html file with @app.route('/')
def index():
    return render_template('index.html')

>Load models and vectorizers saved in file
The following commands can be followed as shown in the figure below


C:\Users\NELSON\Desktop\machine\MLTask\apps\social_eng_app\app.py (MLTask) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
MLTask
.ipynb_checkpoints
apps
social_eng_app
static
templates
index.html
app.py
st_app
data
models
Pipfile
Pipfile.lock
project_workflow.png
README.md
requirements.txt
Social EngineeringTask- Without T
updated_se_dataset.csv
app.py

```python
from flask import Flask,render_template,url_for,request
import joblib,os,re

#init app
app = Flask(__name__)

# load Vectorizer For Social Engineering
count_vectorizer_file = open("static/models/social_eng_vectorizer.pkl","rb")
count_vectorizer = joblib.load(count_vectorizer_file)

# load Vectorizer For Social Engineering
tfidf_vectorizer_file = open("static/models/tfidf_social_eng_vectorizer.pkl","rb")
tfidf_vectorizer = joblib.load(tfidf_vectorizer_file)

se_levels = {0:"not_susceptible",1:"less_susceptible",2:"moderately_susceptible",3:"highly_susceptible"

# For Logistic Regression
logit_model_with_cv = joblib.load(open(os.path.join("static/models/social_eng_logit_model.pkl"),"rb"))
logit_model_with_tfidf = joblib.load(open(os.path.join("static/models/tfidf_social_eng_logit_model.pkl"

# For Naive Bayes
nb_model_with_cv = joblib.load(open(os.path.join("static/models/social_eng_naive_bayes_model.pkl"),"rb"
nb_model_with_tfidf = joblib.load(open(os.path.join("static/models/tfidf_social_eng_naive_bayes_model.p


email_regex = re.compile(r"[\w\.-]+@[\w\.-]+")
phone_num_regex = re.compile(r'\d\d\d.\d\d\d.\d\d\d\d')
phone_num_regex2 = re.compile(r'\d\d\d.\d\d\d.\d\d\d\d')
phone_num_regex_n_iban = re.compile(r'[\+\(]?[1-9][0-9 .\-\(\)]{8,}[0-9]')
iban_regex = re.compile(r'[a-zA-Z]{2}[0-9]{2}[a-zA-Z0-9]{4}[0-9]{7}([a-zA-Z0-9]?){0,16}')

# Function to Extract Email
```

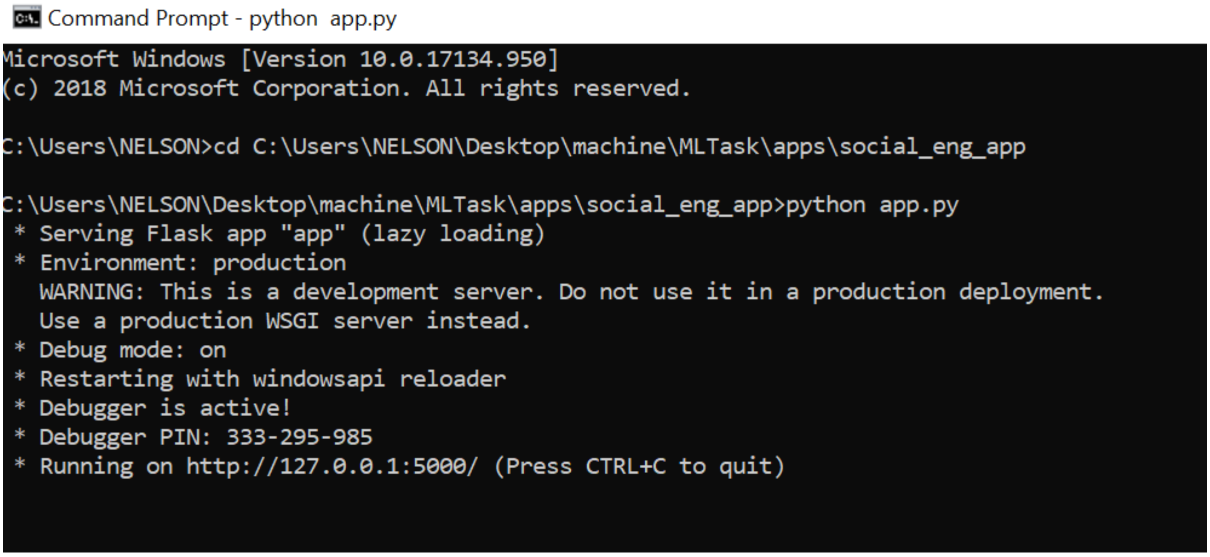
Figure 10 Flask Model integration

K. save the file using ctrl s

SECTION 3

USER GUIDE

a. launch the windows or anaconda command prompt
b. copy the path the saved model cd C:\Users\NELSON\Desktop\machine\MLTask\apps\social_eng_app
c. RUN python app.py
d. Copy the link on the last line after running python app.py



```
Microsoft Windows [Version 10.0.17134.950]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\NELSON>cd C:\Users\NELSON\Desktop\machine\MLTask\apps\social_eng_app

C:\Users\NELSON\Desktop\machine\MLTask\apps\social_eng_app>python app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with windowsapi reloader
 * Debugger is active!
 * Debugger PIN: 333-295-985
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

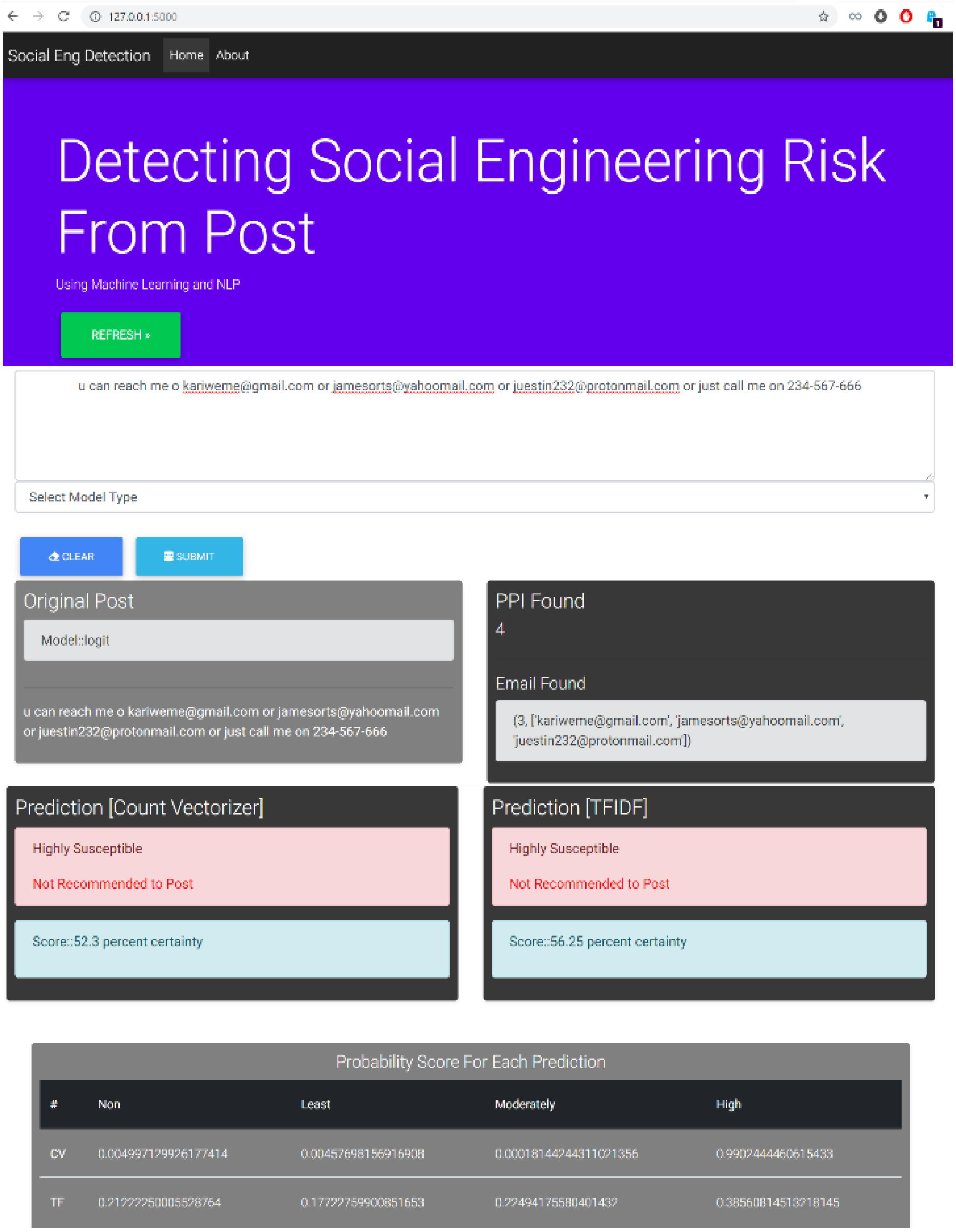
YOU ARE NOW READY TO TEST THE APPLICATION INTEGRATED WITH SE-MLPM

To test the application, follow the following procedure

a. click on the post space and type in a controlled text containing any email address, IBAN or phone number

b. choose the model you want to use in analysis either naïve Bayes or logistic regression

c. SE-MLPM predicts susceptibility of post to social engineering and recommends whether the post should be taken down or not.



Detecting Social Engineering Risk From Post

Using Machine Learning and NLP

u can reach me o kariweme@gmail.com or jamesorts@yahoomail.com or juestin232@protonmail.com or just call me on 234-567-666

Select Model Type

Original Post

Model::logit

u can reach me o kariweme@gmail.com or jamesorts@yahoomail.com or juestin232@protonmail.com or just call me on 234-567-666

PPI Found

4

Email Found

(3, ['kariweme@gmail.com', 'jamesorts@yahoomail.com', 'juestin232@protonmail.com'])

Prediction [Count Vectorizer]

Highly Susceptible

Not Recommended to Post

Score::52.3 percent certainty

Prediction [TFIDF]

Highly Susceptible

Not Recommended to Post

Score::56.25 percent certainty

Probability Score For Each Prediction

#	Non	Least	Moderately	High
CV	0.004997129926177414	0.00457698156916908	0.00018144244311071356	0.9902444460615433
TF	0.21222250005528764	0.17722759900851653	0.22494175580401432	0.38560814513218145

Figure 11 Front end user interface

Model can now continue to learn from subsequent posts and continue predicting susceptibility of post to social engineering attacks

References

[1] M. d. generator, "Mockaroo," 2019. [Online]. Available: https://mockaroo.com/. [Accessed 8 Dec 2019].

[2] O. Ololade, "Towards a Conceptual Model for Mitigating against Social Engineering on the Online Social Network," 2018. [Online]. Available: http://trap.ncirl.ie/3559/1/olabodeololade.pdf. [Accessed 3 Nov 2019].