Configuration Manual

MSc Research Project
MSc in Data Analytics

Alex Greenmount James
Student ID: x21110654

School of Computing
National College of Ireland

Supervisor: Mr Michael Bradford

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Alex Greenmount James
Student ID:	x21110654
Programme:	MSc in Data Analytics
Year:	2022
Module:	MSc Research Project
Supervisor:	Mr Michael Bradford
Submission Due Date:	01/02/2023
Project Title:	Configuration Manual
Word Count:	876
Page Count:	6

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:	
Date:	1st February 2023

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

Alex Greenmount James
x21110654

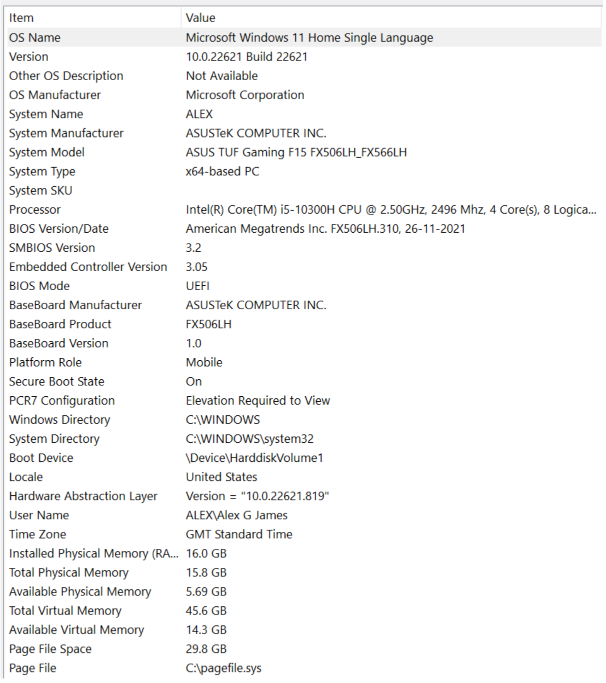
1 Introduction

In this configuration manual, it is explained in detail about the step by step execution procedure or information about the system requirements such as software and hardware specifications, library versions and the storage capacity needed to run/execute the project ”Text Analysis of Russia and Ukraine war”.

2 Software and Hardware Specifications

In this section, the software and hardware specifications of the local machine are detailed.

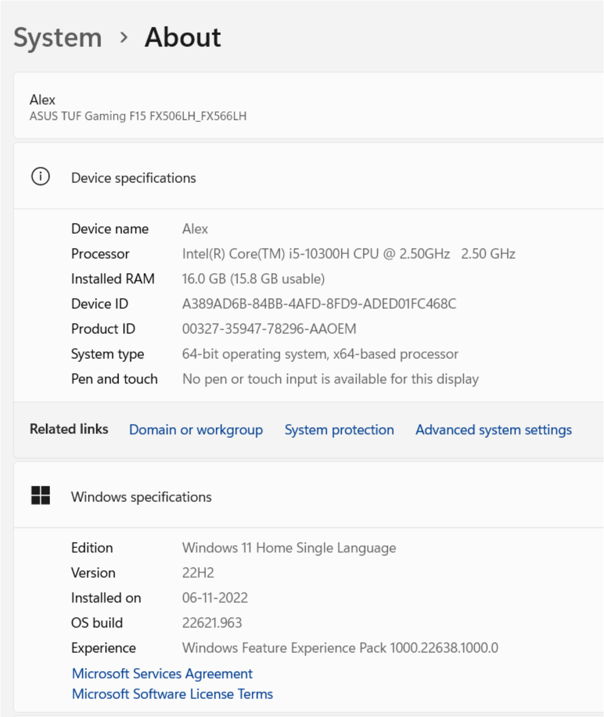
2.1 Hardware Specification of the Local Machine



Item	Value
OS Name	Microsoft Windows 11 Home Single Language
Version	10.0.22621 Build 22621
Other OS Description	Not Available
OS Manufacturer	Microsoft Corporation
System Name	ALEX
System Manufacturer	ASUSTeK COMPUTER INC.
System Model	ASUS TUF Gaming F15 FX506LH_FX566LH
System Type	x64-based PC
System SKU	
Processor	Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz, 2496 Mhz, 4 Core(s), 8 Logica...
BIOS Version/Date	American Megatrends Inc. FX506LH.310, 26-11-2021
SMBIOS Version	3.2
Embedded Controller Version	3.05
BIOS Mode	UEFI
BaseBoard Manufacturer	ASUSTeK COMPUTER INC.
BaseBoard Product	FX506LH
BaseBoard Version	1.0
Platform Role	Mobile
Secure Boot State	On
PCR7 Configuration	Elevation Required to View
Windows Directory	C:\WINDOWS
System Directory	C:\WINDOWS\system32
Boot Device	\Device\HarddiskVolume1
Locale	United States
Hardware Abstraction Layer	Version = "10.0.22621.819"
User Name	ALEX\Alex G James
Time Zone	GMT Standard Time
Installed Physical Memory (RA...	16.0 GB
Total Physical Memory	15.8 GB
Available Physical Memory	5.69 GB
Total Virtual Memory	45.6 GB
Available Virtual Memory	14.3 GB
Page File Space	29.8 GB
Page File	C:\pagefile.sys

Figure 1: System Summary

The above Figure 1 details about the software specifications of the local machine to implement or execute the project.



System › About

Alex
ASUS TUF Gaming F15 FX506LH_FX566LH

Device specifications

Device name	Alex
Processor	Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz 2.50 GHz
Installed RAM	16.0 GB (15.8 GB usable)
Device ID	A389AD6B-84BB-4AFD-8FD9-ADED01FC468C
Product ID	00327-35947-78296-AAOEM
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

Windows specifications

Edition	Windows 11 Home Single Language
Version	22H2
Installed on	06-11-2022
OS build	22621.963
Experience	Windows Feature Experience Pack 1000.22638.1000.0

Microsoft Services Agreement
Microsoft Software License Terms

Figure 2: Hardware and Device Info

The above Figure 2 shows the device specification and windows specification of the local machine.

3 Applications used to run/execute the project

To execute or run the code implemented to complete the project, the applications used are :

- Anaconda version 1.11.0
- Jupyter notebook version 6.4.12
- Google colab python version 3.8.16

4 Python Libraries used

- numpy - 1.21.5
- pandas - 1.4.4
- textblob - 0.17.1
- nltk - 3.7
- keras - 2.11.0

- sklearn - 1.0.2
- matplotlib - 3.5.2
- tensorflow - 2.11.0
- emoji - 2.2.0
- seaborn - 0.11.2
- snscrape - 0.4.3.20220106

Pandas: Pandas is a data analysis library tool used for various data processing. It is a very easily useful library with many plotting and analysis capabilities[1].

Numpy: This is a library used for various numerical calculations in python. This library has many inbuilt functions that quill; help to evaluate the system under complex mathematical conditions[2].

Keras: This is a library associated with machine learning and the neural network it has the capability to implement complex calculations that can support the neural network structure[3].

Sklearn: Sklearn consists of many useful tools like regression classification neural networks that are helpful in catering machine learning algorithms effectively. The capability for this help to do complex machine learning tasks with simple code [4].

Textblob: Textblob is a built-in python library. It is a very useful library for getting the sentiment of the words embedded and the polarity. This is a textual data workable platform. Various processing tools including the lemmatization and tokenization are used with the in-built library of textblob[5].

Matplot: Matplot is used for the visual representation of the data.

5 Coding Files of the Project

- -tweetpull.ipynb: This python coding file contains the code for the tweet collection or scraping from Twitter by using the credentials of developer API using the elevated access. This file is executed in Google Colab.
- -Datacollection.ipynb: This python coding file contains the code for collection or gathering the tweets from Twitter through snscrape library.
- -Preprocessing tweets.ipynb: This python coding file contains the code for the preprocessing or cleaning of the tweets those are gathered.

[1] https://pandas.pydata.org/

[2] https://numpy.org/

[3] https://keras.io/

[4] https://www.javatpoint.com/what-is-sklearn-in-python

[5] https://stackabuse.com/python-for-nlp-introduction-to-the-textblob-library/

- Sentiment.ipynb: This python coding file contains the code for performing the sentiment analysis using NLTK library.

- Random Forest,ipynb: This coding file contains the code for the Random Forest model classification/evaluation.

- ANN.ipynb: This python coding file contains the code for the ANN machine learning model creation.

5.1 Execution format of the Coding Files

1. Collecting/extracting the tweets by executing the tweetpull.ipynb and Datacollection.ipynb files.
2. Preproceesing of the collected tweet data by executing the Preprocessing tweets.ipynb file.
3. Performing the sentiment analysis using NLTK to obtain the polarity, subjectivity and analysis by executing the Sentiment.ipynb file.
4. Performing the machine learning model evaluation/classification with Random Forest model to obtain the results from the test dataset by executing the Random Forest.ipynb file.
5. Training the ANN machine learning model to get the test and training datasets reports by executing the ANN.ipynb file.
Finally, once the results are obtained from the proxy schemes(RF and ANN), the scores and the execution time are compared with the original sentiment analysis engine.

6 Datasets used in the project

- **Zelensky Russia.csv, Ukraine crisis.csv, UkraineNato.csv,Ukraine Russia conflict.csv, Ukraine antiwar, Ukraine war, Russian war, Russian invasion.csv, StandWithUkraine.csv, Russia invade.csv**— All of the mentioned 10 .csv files are the datasets extracted from Twitter for the completion of the project.

- **Common_file_tweets.csv:** All of the 10 datasets obtained by extracting the tweets from Twitter are merged into one dataset, and it is stored in Common_file_tweets.csv.

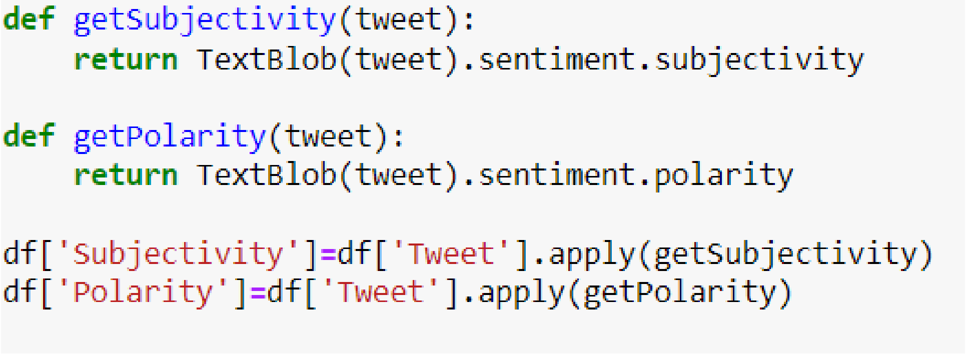
- **formatted main csv.csv:** This datasets contains tweets after preprocessing/cleaned.

- **sentimented tweets.csv:** This datasets contains tweets after the sentiment analysis, having the polarity and subjectivity.

7 Code Snippets

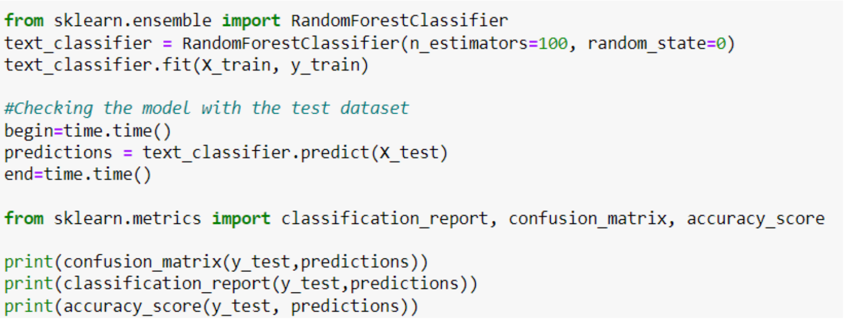
In this section important code snippets are added as screenshots which are used in the project for the implementation.

The Figure 3 shown below is the code snippet for obtaining polarity and subjectivity through the NLTK sentiment analysis engine.



```
def getSubjectivity(tweet):
    return TextBlob(tweet).sentiment.subjectivity

def getPolarity(tweet):
    return TextBlob(tweet).sentiment.polarity

df['Subjectivity']=df['Tweet'].apply(getSubjectivity)
df['Polarity']=df['Tweet'].apply(getPolarity)
```

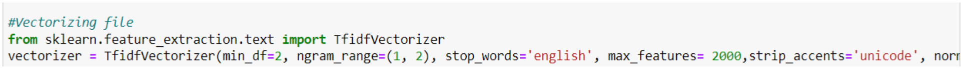
Figure 3: Code snippet for obtaining the polarity and subjectivity through sentiment analysis



```
from sklearn.ensemble import RandomForestClassifier
text_classifier = RandomForestClassifier(n_estimators=100, random_state=0)
text_classifier.fit(X_train, y_train)

#Checking the model with the test dataset
begin=time.time()
predictions = text_classifier.predict(X_test)
end=time.time()

from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

print(confusion_matrix(y_test,predictions))
print(classification_report(y_test,predictions))
print(accuracy_score(y_test, predictions))
```

Figure 4: Checking the RF model with test dataset

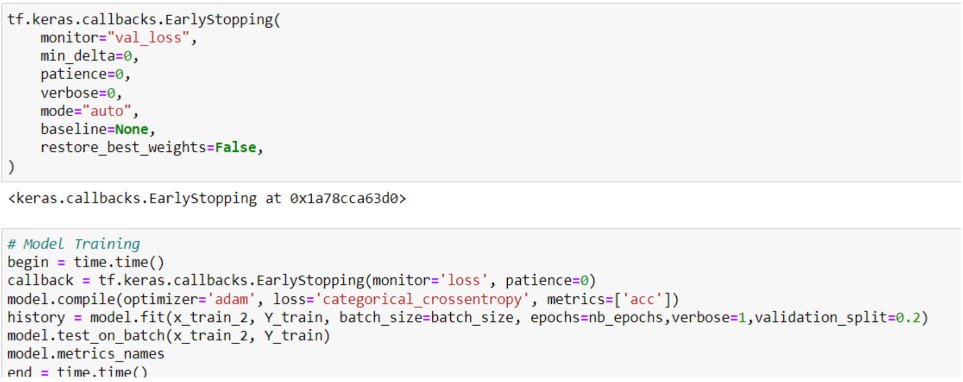
The above Figure 4 is the code snippet used in Random Forest model to perform testing with the test dataset.



```
#Vectorizing file
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(min_df=2, ngram_range=(1, 2), stop_words='english', max_features= 2000,strip_accents='unicode', norm
```

Figure 5: TF-IDF syntax

The Figure 5 above depicts the code snippet for the TF-IDF vectorization.



```
tf.keras.callbacks.EarlyStopping(
    monitor="val_loss",
    min_delta=0,
    patience=0,
    verbose=0,
    mode="auto",
    baseline=None,
    restore_best_weights=False,
)
```

```
<keras.callbacks.EarlyStopping at 0x1a78cca63d0>
```

```
# Model Training
begin = time.time()
callback = tf.keras.callbacks.EarlyStopping(monitor='loss', patience=0)
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['acc'])
history = model.fit(x_train_2, Y_train, batch_size=batch_size, epochs=nb_epochs,verbose=1,validation_split=0.2)
model.test_on_batch(x_train_2, Y_train)
model.metrics_names
end = time.time()
```

Figure 6: ANN model testing

The Figure 6 above shows the code snippet used in the ANN model for training and testing.