National College of Ireland

Configuration Manual

MSc Research Project
Msc in Artificial Intelligence

Anirudh Arora
Student ID: 23151609

School of Computing
National College of Ireland

Supervisor: Victor del Rosal

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Anirudh Arora
Student ID:	23151609
Programme:	Msc in Artificial Intelligence
Year:	2024
Module:	MSc Research Project
Supervisor:	Victor del Rosal
Submission Due Date:	12/12/2024
Project Title:	Configuration Manual
Word Count:	800
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:	Anirudh Arora
Date:	12th December 2024

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

Anirudh Arora
23151609

1 Introduction

It is recommended to follow all the steps proposed in this manual in order to execute the code that is a research project document that should accompany this configuration manual. This document will guide on the requisites both hardware and software that is essential to run the coding programs.

2 Machine Hardware requirements

In order to run the code, the following machine hardware will need to be met for the project to work. The machine that preformed this research configuration is: 16gb RAM, 13th Gen Intel(R) Core(TM) i7-1360P 2.20 GHz, 64-bit OS, Windows 11 Home Single Language.

3 Machine Software requirements

The following software requirements are needed to run the code and are the ones that have been used. Anaconda is used as the environment for the research code. Python was used as the language for this project and version 3.7 respectfully. Jupyter notebook is needed, Microsoft excel is needed as data stored as csv file. Overleaf was used to write the research project report and this manual.

4 Environment set up

Here is where the setting up of the Anaconda environment is done, following these steps will allow the code to run for our research project. The steps are also accompanied by images to better understand the steps taken.

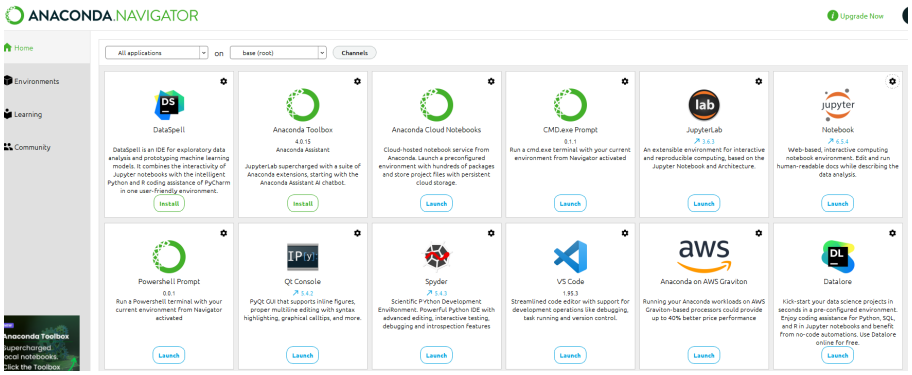


Figure 1: Anaconda Interface

5 Data selection process

The following software requirements are needed to run the code and are the ones that have been used. Anaconda is used as the environment for the research code. Python was used as the language for this project and version 3.7 respectfully. Jupyter notebook is needed, Microsoft excel is needed as data stored as csv file. Microsoft word was used to write the research project report and this manual.

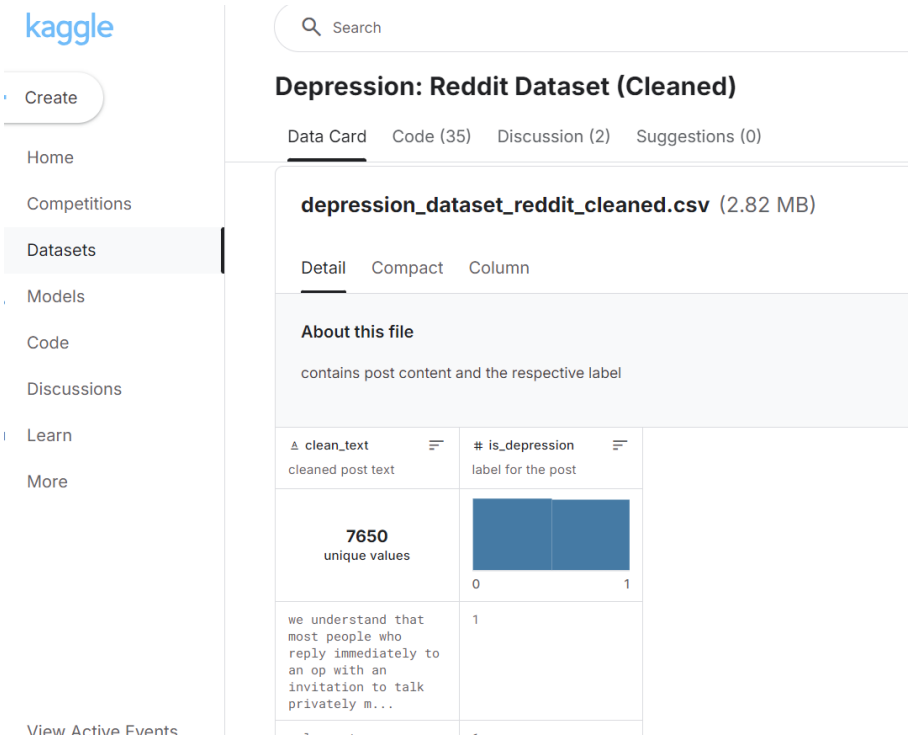


Figure 2: Dataset

6 Install Libraries

For this research the following libraries need to be installed in order for the research to completely work, otherwise results may differ depending on some libraries not being

installed correctly. The below list details all libraries used, with example of ipynb.
1. Pandas
2. Numpy
3. Tensorflow
4. Scikit-learn
5. NLTK
6. Word cloud
7. Seaborn
8. Transformers

In [16]:
```
import numpy as np  # linear algebra
import pandas as pd  # data processing, CSV file I/O
import matplotlib.pyplot as plt
import seaborn as sns
import nltk
from nltk.stem import PorterStemmer, WordNetLemmatizer
from nltk.corpus import stopwords
import re
from tensorflow.keras.layers import LSTM, Dense, Embedding, GRU
from tensorflow.keras.preprocessing.text import one_hot, Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report
from wordcloud import WordCloud
```

Figure 3: Library used

7 Implementation and using the code files

The code files are quite straight forward to use. A quick breakdown of the specifc files that are in the folder. There are 5 jupyter notebook files which contains the codes for 5 different models which are LSTM, Bi-LSTM, BERT, FFNN, GRU. The folder contains the Reddit dataset as well which is read seprately in different models . These models and functions are imported in the relevant jupyter notebooks. This means that the .py files should NOT be run directly because jupyter notebooks are to be used instead. See each jupyter notebook file is numbered and we detail a description below of each.

- **depression dataset reddit cleaned**: The file is a Csv file which contains the dataset named depression_dataset_reddit_cleaned.csv gathered from Kaggle.
- **LSTM:** This file Contains the model code for LSTM to obtain the performance of the model in detecting depressive texts.
- **Bi-LSTM**: This file Contains the model code for Bi-LSTM to obtain the performance of the model in detecting depressive texts.
- **BERT**: This file Contains the model code for BERT to obtain the performance of the model in detecting depressive texts.

- **FFNN:** This file Contains the model code for FFNN to obtain the performance of the model in detecting depressive texts.
- **GRU:** This file Contains the model code for GRU to obtain the performance of the model in detecting depressive texts.

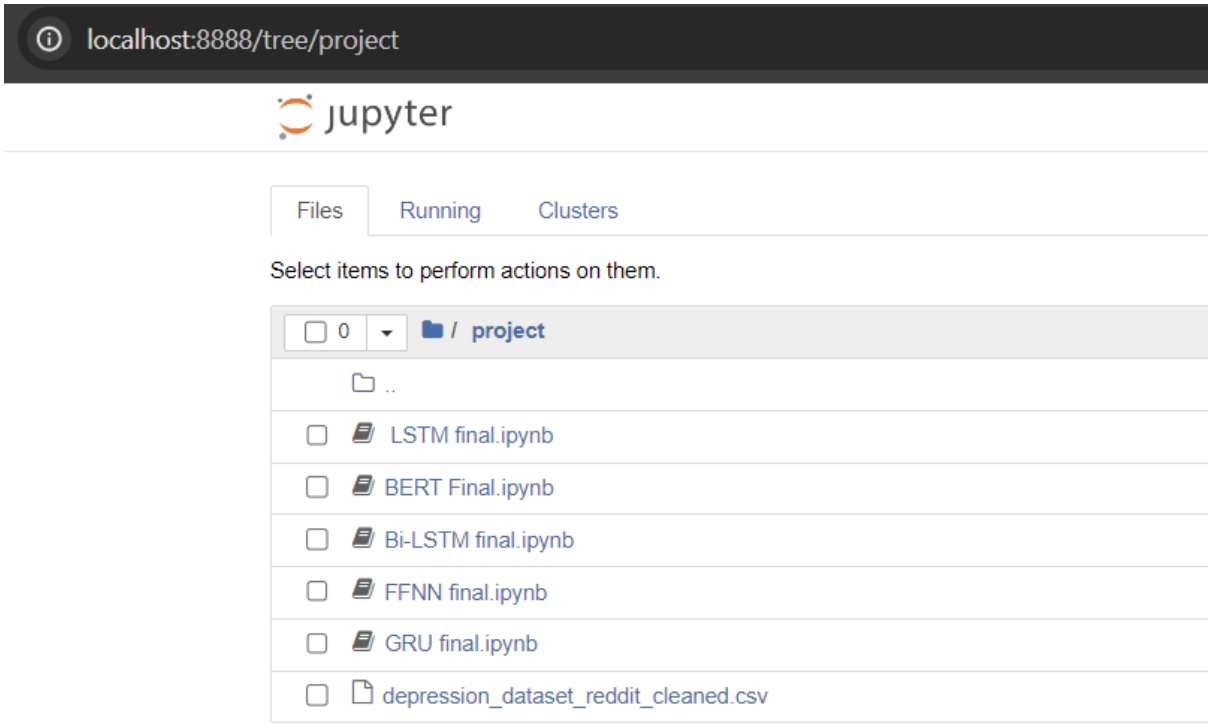


Figure 4: Project Folder containing files.

In order to run each file , open the python notebook files one by one and run the code. Each code contain the data importing, data preprocessing and data visualization code which is common for every model.

In [2]:
```
# Load and preprocess the dataset
df = pd.read_csv('depression_dataset_reddit_cleaned.csv')

# Tokenize the text data
max_features = 18611
tokenizer = Tokenizer(num_words=max_features)
tokenizer.fit_on_texts(df['clean_text'])
sequences = tokenizer.texts_to_sequences(df['clean_text'])

# Pad sequences to ensure uniform length
maxlen = 1844
data = pad_sequences(sequences, maxlen=maxlen)

# Prepare labels
labels = np.array(df['is_depression'])
```

Figure 5: Data Loading and Processing

To have some idea of the data set , it is visualized.

```
[3]:
# Visualize target distribution
plt.figure(figsize=(8, 6))
df['is_depression'].value_counts().plot(kind='bar', color=['skyblue', 'orange'])
plt.title('Distribution of Target Classes')
plt.xlabel('Class')
plt.ylabel('Count')
plt.xticks(ticks=[0, 1], labels=['Not Depressed', 'Depressed'])
plt.show()
```

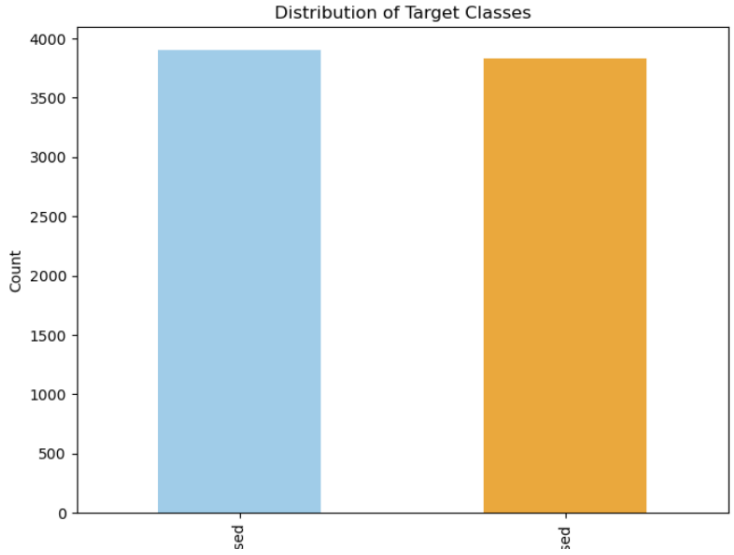
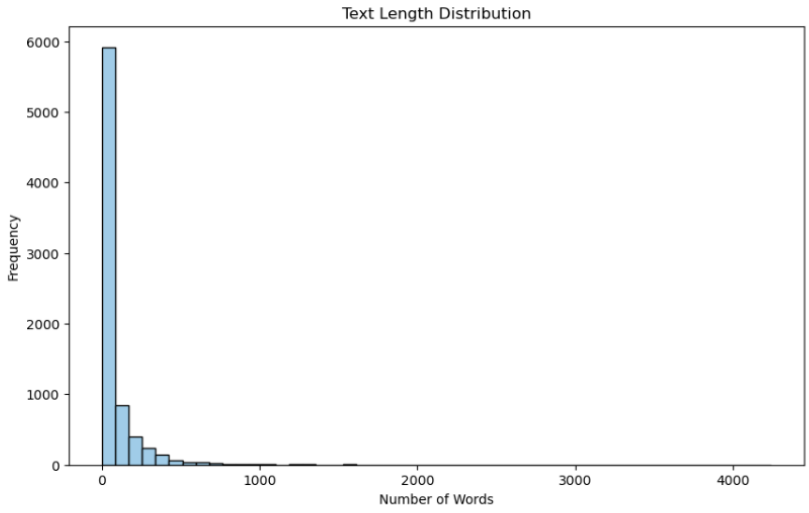


Figure 6: Data Visualization

```
4]:
# Visualize text length distribution
text_lengths = df['clean_text'].apply(lambda x: len(str(x).split()))
plt.figure(figsize=(10, 6))
plt.hist(text_lengths, bins=50, color='skyblue', edgecolor='black')
plt.title('Text Length Distribution')
plt.xlabel('Number of Words')
plt.ylabel('Frequency')
plt.show()

# Print statistics
print(f"Mean text length: {text_lengths.mean():.2f}")
print(f"Median text length: {text_lengths.median():.2f}")
print(f"Max text length: {text_lengths.max()}")
```




```
Mean text length: 74.65
Median text length: 22.00
```

Figure 7: Data Distribution

7.1 Model Training

Each file contains the implementation of the model for training the data set and then validating and testing and generating how the model is performing and what is the accuracy, precision, recall and F1-score. Below is the snapshot of the model implemented as **Feedforward neural network**

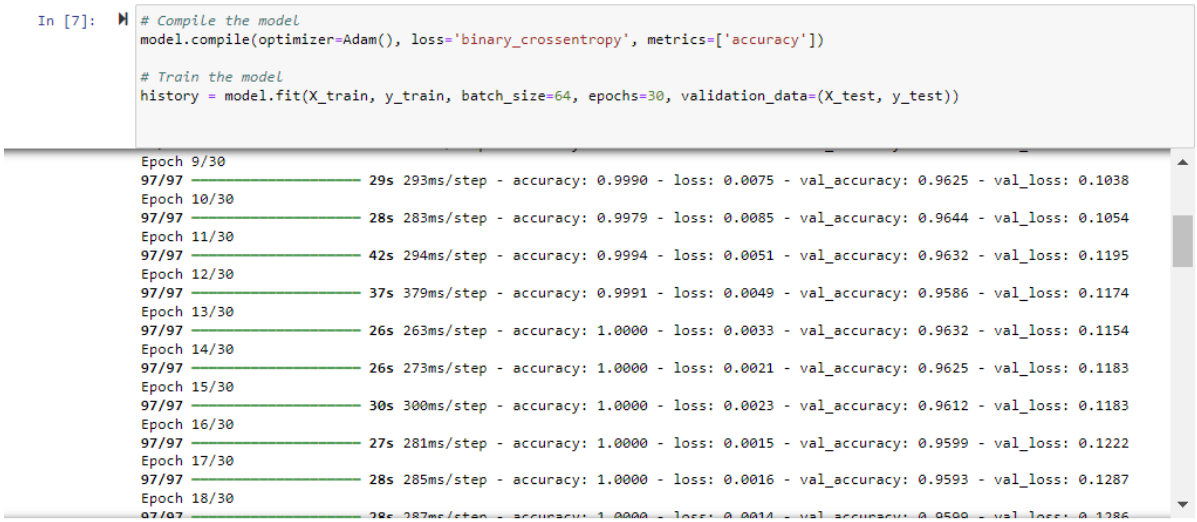


Figure 8: Feed Forward Neural Network Implementation

The same has to be done for the other python notebook (BERT, LSTM, Bi-LSTM, GRU) inorder to obtain the desired result, just click on the Run button and whole code will run on a kernel and will generate the accuracy, precision, recall, F1- score.

References