National College of Ireland

Sentiment Analysis on Covid-19 Vaccination Reviews Using BERT and Comparative Study with LSTM, Vader, and Text blob Models - Configuration Manual

MSc Research Project
Data Analytics

Sourav Ramalingam
Student ID: x20199911

School of Computing
National College of Ireland

Supervisor: Mr. Taimur Hafeez

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: MR. Sourav Ramalingam

Student ID: x20199911

Programme: Data Analytics **Year:** 2022

Module: MSc Research Project

Lecturer: Mr. Taimur Hafeez

Submission Due Date: 15/12/2022

Project Title: Sentiment Analysis on Covid-19 Vaccination Reviews Using BERT and Comparative Study with LSTM, Vader, and Text blob Models.

Word Count: ……1142…… **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.

Signature: Sourav Ramalingam……

Date: 15/12/2022……

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

Sourav Ramalingam
Student ID: x20199911

1. Introduction

In the manual configuration document, the system specification that has been used for the application development is documented. Documenting the configuration details would help to set up the code in any other system for evaluation, making enhancements and so on. On top of the system configuration, the document also contains the packages that are used in the chosen programming language. These details are very important as multiple versions are available for the same package and the combination of the right ones needs to be used.

The document also highlights the steps like pre-processing the content, applying the model and so on.

2. System Configuration

In this section, the hardware, software, and programming configuration that is used for model development are documented.

2.1 Hardware & Base Specification:

Operating System	Windows 11 Home Edition
Installed RAM	8.00 GB
Processor	Intel Core i5 CPU @2.50 GHz
System Type	64-Bit Operating System
Programming Language	Python Programming
Package Management	PIP
Development Environment	PyCharm Free Community Edition

2.2 Software Specification:

The application is developed using the Python programming language and the base python package version used is 3.10.x.

The application also uses the below components:

- HTML for the web user interface development.
- Bootstrap CSS for designing the web page more easily.
- jQuery version 3.4.1 for the client-side scripting operations.
- Roboto Slab Font from online CDN.
- Python Programming for the Core Model Development.
- Python Flask module for developing the WEB API URL access to interact from the client-side HTML pages.

2.3 Package Specification:

The below-mentioned packages are used in Python programming for application development. The packages are downloaded from the package management system called 'PIP'.

Package Name	Version	Package Description
NLTK	3.7	Package used for natural language processing.
Word2number	1.1	Package used to convert the numbers in words to numeric.
Pandas	1.5.1	Package provides a fast and flexible way of working with the data.
TensorFlow	2.10.0	Package for open-source machine learning framework.
Gensim	4.2.0	Package used for document indexing and similarity retrieval from the huge text.
Keras	2.10.0	An application programming interface (API) developed by Google for implementing neural networks.
Numpy	1.23.4	Package used for performing powerful operations on the array object with multiple dimensions.
Scikit-learn	1.1.3	Package that is used to perform machine learning tasks which are built on top of SciPy.
VaderSentiment	3.3.2	Package to consume the VADER (Valence Aware Dictionary and sEntiment Reasoner) which is a lexicon and rule-based sentiment analysis tool.
TextBlob	0.17.1	Library that processes the textual data and find the sentiment.
Torch	1.13.0	Package that provides features like tensor computation and building deep neural networks.
Transformers	4.23.1	Package that has pretrained models to perform tasks on text, vision, and audio data.
Flask	2.2.2	It is a lightweight WSGI web application framework, used for API development.

3. Data Source

The vaccination-related tweets extracted from the Twitter feed are used for the model development. The source has N Number of columns, anyways for finding the sentiment score, the user review text and the rating would be taken into consideration. The rest of the fields are dropped while processing the same in the python coding.

The dataset has total records of 1,25,906.

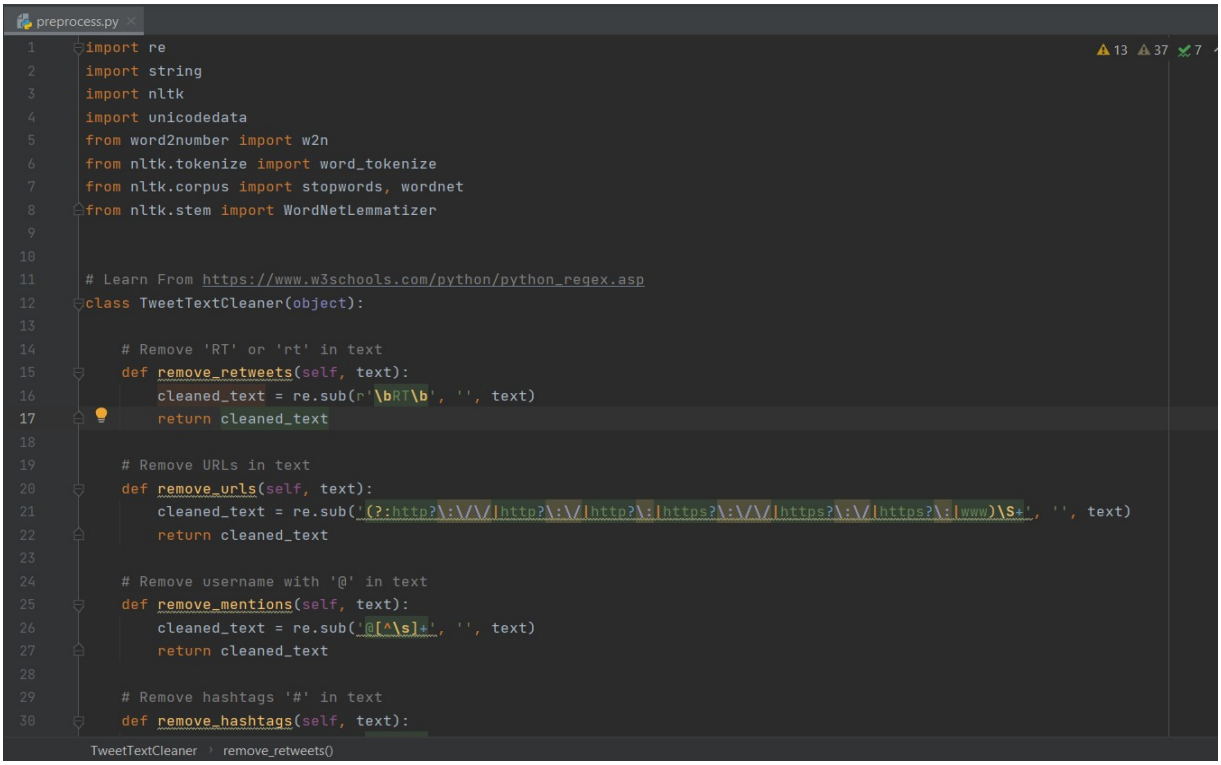
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	id	user_name	user_locat	user_descr	user_creat	user_follo	user_frien	user_favo	user_verifi	date	text	hashtags	source
2	1.34E+18	Rachel Rol	La Crescer	Aggregato	########	405	1692	3247	FALSE	########	Same folks	['PfizerBio	Twitter fo
3	1.34E+18	Albert Fon	San Franci	Marketing	########	834	666	178	FALSE	########	While the world has b		Twitter W
4	1.34E+18	eliðŸ‡±ðŸ‡	Your Bed	heil, hydra	########	10	88	155	FALSE	########	#coronavii	['coronavi	Twitter fo
5	1.34E+18	Charles Ad	Vancouver	Hosting "C	########	49165	3933	21853	TRUE	########	Facts are immutable,		Twitter W
6	1.34E+18	Citizen News Channel		Citizen Ne	########	152	580	1473	FALSE	########	Explain to	['whereare	Twitter fo
7	1.34E+18	Dee	Birminghar	Gastroente	########	105	108	106	FALSE	########	Does anyone have an		Twitter fo
8	1.34E+18	Gunther Fe	Austria, UK	End North	########	2731	5001	69344	FALSE	########	it is a bit sa	['vaccinati	Twitter W
9	1.34E+18	Dr.Krutika Kuppalli		ID, Global	########	21924	593	7815	TRUE	########	There	['BidenHar	Twitter fo
10	1.34E+18	Erin Despas		Designing&	########	887	1515	9639	FALSE	########	Covid	['CovidVac	Twitter W
11	1.34E+18	Ch.Amjad .	Islamabad	#ProudPa	########	671	2368	20469	FALSE	########	#CovidVa	['CovidVac	Twitter W
12	1.34E+18	Tamer Yaz	Turkey-Isr	Im	########	1302	78	339	FALSE	########	while	['PfizerBio	Twitter W
13	1.34E+18	VoiceM		campaigne	########	2	25	20	FALSE	########	@cnnbrk #	['COVID19	Twitter W
14	1.34E+18	WION	India	#WION: W	########	292510	91	7531	TRUE	########	The agency also relea		TweetDe
15	1.34E+18	Dr.Krutika Kuppalli		ID, Global	########	21924	593	7815	TRUE	########	For all the	['PfizerBio	Twitter fo
16	1.34E+18	Opoyi		High-qualit	########	10332	49	16	FALSE	########	"Expect 145 sites acro		TweetDe
17	1.34E+18	City A.M.	London, Er	London's b	########	66224	603	771	TRUE	########	Trump	['vaccine']	Twitter fo
18	1.34E+18	STOPCOM	Global	'Trust' is n	########	406	176	479	FALSE	########	UPDATED	['YellowFe	Twitter W
19	1.34E+18	ILKHA	TÃ¼rkiye	Official Tw	########	4056	6	3	TRUE	########	Coronaviru	['Iran', 'co	TweetDe
20	1.34E+18	Braderz73	Bristol, UK	One of	########	6430	6292	45007	FALSE	########	.@Pfizer w	['CovidVac	Twitter fo
21	1.34E+18	Alex Vie	Los Angele	Marine vet	########	125	442	5401	FALSE	########	The trump	['COVIDIO	Twitter fo

4. Code Setup – Step-by-Step

- Download the source code and place it in a local path under an empty folder.
- Find the /dataset folder and place the downloaded dataset file from the provided link if the folder is empty.
- Ensure Python 3.10.X is installed.
- Ensure PyCharm Community Version IDE or any similar one is installed.
- Open the source code with IDE and it would prompt to create a Virtual Environment, create the same.
- To activate the virtual environment, the below code needs to be executed in the Terminal:
    - Cd env/Scripts activate
- Install the required packages with the below comment:
    - Pip3 install -r requirements.txt
- Execute the below command to initiate the web app and interact with the UI by providing sample inputs and viewing the results:
    - Python -m flask run

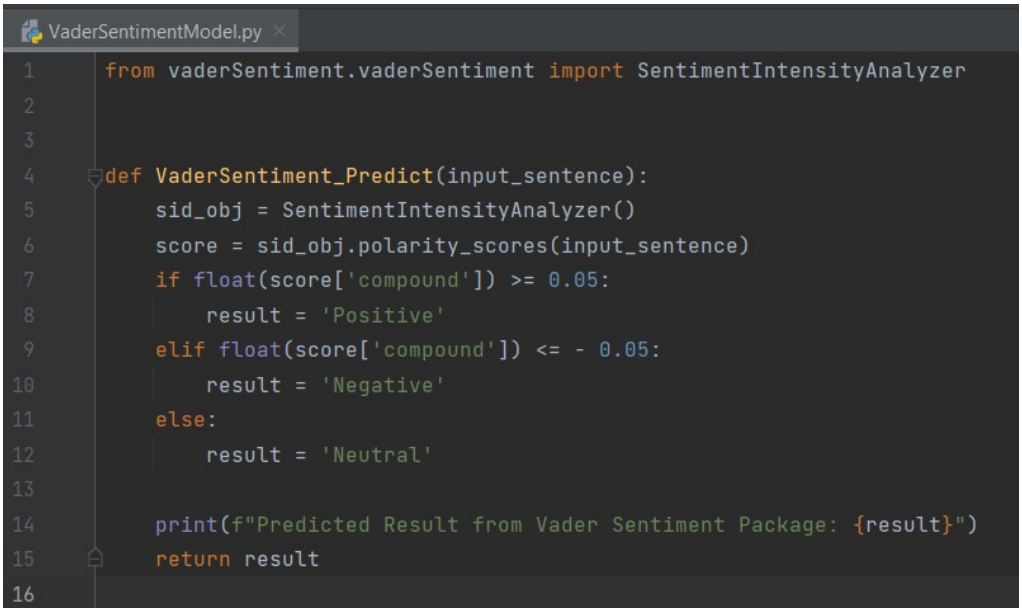
5. Source Code Modules

The below code snippet is to perform the pre-processing on the tweet text before it is taken for processing by the model.



```python
import re
import string
import nltk
import unicodedata
from word2number import w2n
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords, wordnet
from nltk.stem import WordNetLemmatizer


# Learn From https://www.w3schools.com/python/python_regex.asp
class TweetTextCleaner(object):

    # Remove 'RT' or 'rt' in text
    def remove_retweets(self, text):
        cleaned_text = re.sub(r'\bRT\b', '', text)
        return cleaned_text

    # Remove URLs in text
    def remove_urls(self, text):
        cleaned_text = re.sub('(?:http?\:\/\/|http?\:\/|http?\:|https?\:\/\/|https?\:\/|https?\:|www)\S+', '', text)
        return cleaned_text

    # Remove username with '@' in text
    def remove_mentions(self, text):
        cleaned_text = re.sub('@[^\s]+', '', text)
        return cleaned_text

    # Remove hashtags '#' in text
    def remove_hashtags(self, text):
```

The below code snippet is to calculate the sentiment score using the Vader Sentiment Package.



```python
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer


def VaderSentiment_Predict(input_sentence):
    sid_obj = SentimentIntensityAnalyzer()
    score = sid_obj.polarity_scores(input_sentence)
    if float(score['compound']) >= 0.05:
        result = 'Positive'
    elif float(score['compound']) <= - 0.05:
        result = 'Negative'
    else:
        result = 'Neutral'

    print(f"Predicted Result from Vader Sentiment Package: {result}")
    return result

```

The below code snippet is to calculate the sentiment score using the TextBlob Sentiment Package.

TextBlobModel.py

```python
from textblob import TextBlob


def TextBlob_Predict(input_sentence):
    data = TextBlob(input_sentence)
    result = ''
    if data.sentiment.polarity < 0:
        result = 'Negative'
    elif data.sentiment.polarity == 0:
        result = 'Neutral'
    elif data.sentiment.polarity > 0:
        result = 'Positive'

    print(f"Predicted Result from TextBlog Package: {result}")
    return result

```

The below code snippet is the LSTM Sentiment Analysis Model that performs both the model training and prediction on based of the value of the mode variable: ‘test/train’.

SentimentAnalysis_LSTM.py

```python
        model1.add(layers.Embedding(max_words, 20))
        model1.add(layers.LSTM(15, dropout=0.5))
        model1.add(layers.Dense(3, activation='softmax'))

        model1.compile(optimizer='rmsprop', loss='categorical_crossentropy', metrics=['accuracy'])
        checkpoint1 = ModelCheckpoint("model_weight/LSTM_Model.hdf5", monitor='val_accuracy', verbose=1,
                                      save_best_only=True, mode='auto',
                                      period=1, save_weights_only=False)
        model1.fit(X_train, y_train, epochs=10, validation_data=(X_test, y_test), callbacks=[checkpoint1])

        best_model = keras.models.load_model("model_weight/LSTM_Model.hdf5")
        model_trained = True
        lstmpredict = model1.predict(X_test)
        print(lstmpredict)

        test_loss, test_acc = best_model.evaluate(X_test, y_test, verbose=2)
        print('Model accuracy: ', test_acc)
        print(classification_report(np.argmax(y_test, axis=1), np.argmax(lstmpredict, axis=1)))
        return test_acc

    elif mode == 'test':
        if not model_trained:
            train_predict('train')
        print('testing inputs')
        sentiment = ['Neutral', 'Negative', 'Positive']
        sequence = tokenizer.texts_to_sequences([input])
        test = pad_sequences(sequence, maxlen=max_len)
        result = sentiment[np.around(best_model.predict(test), decimals=0).argmax(axis=1)[0]]
        print(f"Predicted Result from Long Short Term Memory (LSTM) Sentiment Analysis Module: {result}")
        return result
```

The below code snippet is the BERT Model training code:

BERT_ModelTraining.py

```
    best_accuracy = 0

    for epoch in tqdm(range(EPOCHS)):
        print(f"Epoch {epoch + 1}/{EPOCHS}")
        print("-" * 10)

        train_acc, train_loss = train_epoch(
            model, train_data_loader, loss_fn, optimizer, device, len(df_train)
        )

        print(f"Epoch: {epoch}, Train loss: {train_loss}, accuracy: {train_acc}")

        val_acc, val_loss = eval_model(
            model, val_data_loader, loss_fn, device, len(df_val)
        )

        print(f"Epoch: {epoch}, Val loss: {val_loss}, accuracy: {val_acc}")

        history["train_acc"].append(train_acc)
        history["train_loss"].append(train_loss)
        history["val_acc"].append(val_acc)
        history["val_loss"].append(val_loss)

        if val_acc > best_accuracy:
            torch.save(model.state_dict(), "model_weight/BERT_Model.hdf5")
            best_accuracy = val_acc

    print("Training completed")

```

The below code snippet is the BERT Model Prediction code:

BERT_ModelPredict.py

```
import keras
import numpy as np
import pickle
from keras_preprocessing.sequence import pad_sequences


def predict(input_sentence=''):
    max_len = 200
    with open('model_weight/tokenizer.pickle', 'rb') as handle:
        tokenizer = pickle.load(handle)
    data = [input_sentence]
    tokenizer.fit_on_texts(data)
    best_model = keras.models.load_model("model_weight/BERT_Model.hdf5")
    sentiment = ['Neutral', 'Negative', 'Positive']
    sequence = tokenizer.texts_to_sequences(data)
    test = pad_sequences(sequence, maxlen=max_len)
    result = sentiment[np.around(best_model.predict(test), decimals=0).argmax(axis=1)[0]]
    print(f"Predicted Result from BERT Sentiment Analysis Module: {result}")
    return result

```

The below code snippet is for exposing the Python Flask API:

app.py

```
from LSTM import SentimentAnalysis_LSTM

app = Flask(__name__)
app.secret_key = "secret key"


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


@app.route('/modelexec', methods=['GET', 'POST'])
def model_execution():
    if request.method == 'GET':
        input_Text = request.args.get("input_Text")
        print(input_Text)
        result = {
            "textblob": TextBlobModel.TextBlob_Predict(input_Text),
            "vader": VaderSentimentModel.VaderSentiment_Predict(input_Text),
            "bert": BERT_ModelPredict.predict(input_Text),
            "lstm": SentimentAnalysis_LSTM.train_predict('test', input_Text)
        }
        return result;


if __name__ == '__main__':
    app.run(debug=True)
```

Related Links:

- Python Download Link: https://www.python.org/downloads/release/python-3100/
- Pip: https://pypi.org/project/pip/
- PyCharm IDE Download: https://www.jetbrains.com/pycharm/download/
- Covid-19 Vaccination Dataset: https://www.kaggle.com/datasets/gpreda/all-covid19-vaccines-tweets
- NLTK - https://pypi.org/project/nltk/
- Word2number - https://pypi.org/project/word2number/
- Pandas - https://pypi.org/project/pandas/
- TensorFlow - https://www.tensorflow.org/install/pip
- Gensim - https://pypi.org/project/gensim/
- Keras - https://pypi.org/project/keras/
- NumPy - https://pypi.org/project/numpy/

- Scikit-learn - https://pypi.org/project/scikit-learn/
- Vader Sentiment - https://pypi.org/project/vaderSentiment/
- Text Blob - https://pypi.org/project/textblob/
- Torch - https://pypi.org/project/torch/
- Transformers - https://pypi.org/project/transformers/
- Flask - https://pypi.org/project/Flask/