# CSE4037 – DEEP LEARNING

## J Component Report

**A project report titled**

# Emotion Classification and Personality Prediction


*By*

| | |
|---|---|
| Reg. No:19MIA1004 | HARINI GOKULRAM NAIDU |
| Reg. No:19MIA1006 | SHIVANI GOKULRAM NAIDU |
| Reg.No:19MIA1050 | HARSHINI K AIYYER |
| Reg.No:19MIA1059 | ANNUGRAHA S |


MASTER OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

*Submitted to*

## Dr.R.Rajalakshmi

**School of Computer Science and Engineering**


VIT®

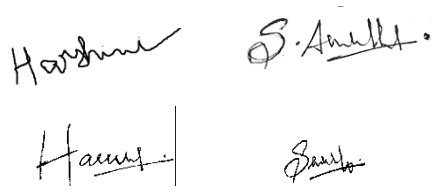
Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

*April 2022*

## DECLARATION BY THE CANDIDATE

I hereby declare that the report titled "**Emotion Classification and Personality Prediction"** submitted by me to VIT Chennai is a record of bona-fide work undertaken by me under the supervision of **Dr. R. Rajalakshmi, Associate Professor, SCOPE, Vellore Institute of Technology, Chennai.**



Signature of the Candidate(s)

# ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr. R. Rajalakshmi,** School of Computer Science and Engineering for her consistent encouragement and valuable guidance offered to us throughout the course of the project work.

We are extremely grateful to **Dr. R. Ganesan, Dean,** School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology, Chennai, for extending the facilities of the School towards our project and for his unstinting support.

We express our thanks to our **Head of the Department** for his support throughout the course of this project.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the courses.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

# BONAFIDE CERTIFICATE

Certified that this project report entitled "**Emotion Classification and Personality Prediction"** is a bona-fide work of **HARINI (19MIA1004), SHIVANI (19MIA1006), HARSHINI (19MIA1050), ANNUGRAHA (19MIA1059)** carried out the "*Emotion Classification and Personality Prediction*"-Project work under my supervision and guidance for CSE4037 – DEEP LEARNING.

**Dr.R. Rajalakshmi**

SCOPE

# TABLE OF CONTENTS

## ABSTRACT


## Track 2: Emotion Classification (EMO)

The goal of this project is to determine from the essay what emotion is felt. Natural language processing, deep learning and machine learning algorithms are used to predict the emotion. The classes are joy, neutral, surprise, fear, sadness, disgust and anger. The evaluation metric will be accuracy, macro F1-score, micro F1-score, micro precision, micro recall, macro precision and macro recall.

Some of the existing problems are that the use of text-based emotional detection has not been adequately explored for certain critical or life-saving applications. These areas include crime detection and mitigation in which analyzing messages of victims in order to identify threatening words, analysis of patient messages in order to determine depression levels of patients so that timely support can be offered, etc. Identifying from the enormous amount of text data generated on a daily basis by people, undertaking research activities around the listed application areas among other similar life-saving applications would enhance and sustain the relevance of text based emotional detection.

## Track 3: Personality Prediction (PER)

The project is based on identifying the personality of an individual who wrote a text using the big 5 model. The personality of a human plays a major role in his personal and professional life.

The Big Five model is also known as the Five-Factor Model (FFM) and OCEAN model was developed in the early 1980s according to many psychological theories. When the

statistical analysis is applied to personality survey data, some words used to describe the person and these words give a summary of the overall character or personality of the person accurately.

Open to Experience: It involves various dimensions, like imagination, sensitivity, attentiveness, preference to variety, and curiosity.

Conscientiousness: This trait is used to describe the carefulness and diligence of the person. It is the quality that describes how organized and efficient a person is.

Extraversion: It is the trait that describes how the best candidates can interact with people that is how good are his/her social skills.

Agreeableness: It is a quality that analyses the individual behaviour based on the generosity, sympathy, cooperativeness and ability to adjust with people.

Stability: This trait usually describes a person to have mood swings and has extreme expressive power.

A major issue with people is that they do not know which job role would best suit their personality and hence there is inefficient performance and the employees are not happy in the work places. This leads to unhappy employees as well as employers.

This test is used to measure a person's most important personality characteristics, and help him to understand which roles suit him best. Nowadays, many organizations have also started shortlisting the candidates based on their personality as this increase the efficiency of the work because the person is working in what he is good at than what he is forced to do.

We will be using NLP and deep learning models in order to proceed with the project.

# INTRODUCTION

# EMOTION CLASSIFICATION

Sentiment analysis is closely related to emotion detection. In computer science, text categorization in emotional states is known as sentiment analysis or emotion detection. Text can trigger emotions when someone who reads the text and also can reflect or express the emotional state of the person who wrote it. Humans have the power to feel different types of emotions because human life is filled with many emotions. Happy (joy), fear, anger, and sadness are some of the emotional states that can be found in everyday life. Deep learning uses deep neural networks to study input data that can be a good representation, which can then perform a specific task. Also, sentiment analysis (positive or negative) using deep learning has been showed to have a better accuracy compared to the traditional machine learning such as Naïve Bayes (NB) and SVM. LSTM can be used to carry out sentiment analysis for classifying sentiment into positive and negative sentiments [9]. LSTM itself has various kinds of architectures such as Nested LSTM, Bi-LSTM, Gated Recurrent Unit (GRU), and Backpropagation Through Time (BPTT). Therefore, we are interested in examining the emotion detection found in the text by using deep learning, especially Bi-LSTM.

# PERSONALITY PREDICTION:

In recent years, information growth has proliferated in accelerating pace in line with the advent of social media especially in the form of textual data types. According to

the Social Media Trend report published, there are 3.8 billion active users of social media in the world as of January 2020, with a projected increase of 9.2% of users each year. Often, people use social media to express themselves on certain issues related to their lives and family well beings, psychology, financial issues, interaction with societies and environment, as well as politics. In some cases, these expressions can be used to characterize the individual behaviour and personality. In fact, earlier studies demonstrate that there is a strong correlation between user personalities and their online behaviour on social media. Some examples of applications that can take advantage from the user personality information include recruitment systems, personal counselling systems, online marketing, personal recommendation systems, and bank credit scoring systems to name a few. In this project we used deep neural networks using regression to predict the personality based on the given Big Five personality traits. We also implemented gradient boosting regressor, LightGBM and XGBoost to show the comparison.

## LITERATURE SURVEY

### 1) A REVIEW ON EMOTION DETECTION AND RECOGNITION FROM TEXT USING NATURAL LANGUAGE PROCESSING

**Prof. Hardik S. Jayswal, Dhruvi D. Gosai, Himangini J. Gohil**

The Ekman model includes happy or joy, sad, disgust, anger, fear and surprise as basic emotion. Emotion Detection from textual source can be done using concepts of Natural Language Processing. In this paper emotion detection using NLP and its naive algorithm are described. In this approach, we are classifying the input text into different emotions by finding the emotional content from the given

English text. Some of the steps in this paper are text processing, defining dictionaries of basic six expressions, tokenization and POS tagging, increment and decrement of sentimental measure and Inverters and polarity flips.

## 2)EMOTXT: A TOOLKIT FOR EMOTION RECOGNITION FROM TEXT

**Fabio Calefato, Filippo Lanubile, Nicole Novielli University of Bari "Aldo Moro"**

In this paper, they developed EmoTxt, the first open-source toolkit for emotion recognition from text. The toolkit can be used by researchers for detecting emotions (love, joy, anger, sadness, fear, and surprise) from input text as well as for training a custom emotion classifier from scratch, based on manually annotated data. The system was completely developed in Java and distributed under the MIT open-source license. They built a gold standard dataset composed of 4,800 posts (question, answer, and comments) from Stack Overflow. With the toolkit, they released the classification models trained on gold standard datasets , which can be used for emotion detection from text. EmoTxt identifies emotions in an input corpus provided as a CSV file, with one text per line, preceded by a unique identifier. The output is a CSV file containing the text id and the predicted label for each item of the input collection. They trained the EmoTxt classification models in a supervised machine learning setting using Support Vector Machines (SVM). Linear SVM is a state-of-the-art learning technique for such high-dimensional sparse datasets with a large number of items and a large number of features. They trained EmoTxt as a suite of six binary classifiers, which predict the presence/absence of each emotion in the input text.

## 3)BILSTM-ATTENTION AND LSTM-ATTENTION VIA SOFT VOTING IN EMOTION CLASSIFICATION

**Qimin Zhou, Zhengxin Zhang, Hao Wu* School of Information Science and Engineering, Yunnan University Chenggong Campus, Kunming, P.R. China**

The goal of this paper is to classify the emotions of excluded words in tweets into six different classes: sad, joy, disgust, surprise, anger and fear. For this they implemented a BiLSTM architecture with attention mechanism (BiLSTM-Attention) and a LSTM architecture with attention mechanism (LSTM-

Attention), and tried different dropout rates(0.1 to 0.6 with a step of 0.1.) based on these two models. Then exploited an ensemble (soft voting) of these methods to give the final prediction which improved the model performance significantly compared with the baseline model. They implemented this on Keras with a Tensorflow backend.

The BiLSTM-Attention (macro F1 score = 0.662) model performs slightly better than the LSTMAttention (0.661) model because BiLSTM can learn more features than LSTM.

After ensembling the LSTM-Attention model and the BiLSTM-Attention model with different dropout rates, the macro F1 score reaches to 0.685.

# 4)A DEEP LEARNING BASED APPROACH FOR MULTI-LABEL EMOTION CLASSIFICATION IN TWEETS

**What the paper is doing –**

A novel, attentive deep learning system, which we call Binary Neural Network (BNet), which works on the new transformation method and a xy pair-set transformation method.

**Criticisms -** At a holistic level, the positive and negative correlations between the different emotions might speak for itself, the authors feel that an in-depth analysis might have conflicts with the same. There could be a lot of cases where a person could have conflicting emotions (you can be sad yet be optimistic) and yet the matrix has shown negative correlations for the same. Users can have different views and methods of writing for a given incident. The measures undertaken in the paper doesn't account for the sarcastic tweets that are quite frequent on social media nowadays.

# 5)AN EVALUATION OF THE SHORT CONSEQUENCES OF USING SHORT MEASURES OF THE BIG FIVE PERSONALITY TRAITS

**What the paper is doing**

This paper aims to examine this issue by comparing the criterion-related variance captured by scores on eight publicly available shortened scales of the Big Five personality traits. Eight scenarios were taken for the source of study where several – item metrics were taken into consideration.

**Criticisms**

The statistic used here is Standard Deviation and Mean which could be a disadvantage as SD assumes all distributions to be of equal weights The paper seems to put forth itself as a solution for the test subjects from taking long and exhausting tests, by conducting single-item tests meanwhile contradicting itself – putting up results deduced from eight-item psychological tests.

# 6)BIG FIVE PERSONALITY DETECTION USING DEEP CONVOLUTIONAL NEURAL NETWORKS

**Waiel Tinwala, Shristi Rauniyar**

We have used the Big Five Model often known as the five-factor model or OCEAN model. Document- level feature extraction has been performed. The processed data has been fed into a deep convolutional network and a binary

classifier has been used to classify the presence or absence of the personality trait. Holdout method has been used to evaluate the model, and the F1 score has been used as the performance metric.The proposed model had used k-fold cross-validation technique (k=10) whereas they have used hold-out method because of limited resources and yet have achieved better results.Sigmoid and Tanh are non-linear activation functions whereas ReLU is a linear function. They have trained the model using these three activations and compared the results. This model performs better than the state-of-the-art model.

## 7)PREDICTING BIG FIVE PERSONALITY TRAITS OF MICROBLOG USERS

**Shuotian Bai, Bibo Hao, Ang Li, Sha Yuan, Rui Gao, Tingshao Zhu**

The paper analyzes the personality based on the big-five theory. It proposes an incremental regression model to prove the reliability of the dataset and the multi-task regression model to improve the predicting accuracy. It tries to find the associated modes of users' personality and network characteristics and through different machine learning algorithms, computational models of personality based on network characteristics are founded. In order to test the performance of the different models, 5-fold cross validation for training and Mean Absolute Error (MAE) is used as the assessment criteria . correlation analysis is done using pearson correlation Agreeableness, conscientiousness, extraversion, and openness pairwise show a significant positive correlation . However, neuroticism is significantly negatively correlated with the other four dimension. You find that the average of MAE is least for multitask regression with a value of 0.1384. The reliability of the dataset is verified, and it shows how personality influence and reflect the online behaviors.

**LIMITATIONS**

A larger dataset could have been used(here 444 users)

User network characteristics could have been extracted(to find social attitude , behavioural pattern

# 8)HUMAN EMOTION DETECTION AND CLASSIFICATION IN TEXT MINING

**S. Ambar , S. Jan and F. Khan**

This research classifies the emotion of the text in four categories Happy, Sad, Anger, Other. Keyword based approach is used with the aim that handling the contradictory conjunction can improve the textual emotion classification. Here we have seen that handling the contradictory conjunction or contrasting sentences can improves the accuracy from 0.5928 to 0.5964. The paper suggests that removing the cause of emotion can improve the efficiency of textual emotion classifier. The accuracy of the classifier increases when tested on text with cause from 0.5636 to 0.5784.

**LIMITATIONS**

Cause is not detected accurately because the cause detection algorithm incorrectly detects and removes essential text from sentence which do not have cause with the

# 9)A NEURAL NETWORK APPROACH TO PERSONALITY PREDICTION BASED ON THE BIG-FIVE MODEL

**Mayuri pundlik Kalghatgi manjula ramannavar Dr. nandini s. sidnal**

In this paper they predict the traits of an individual using the group of tweets posted by him. The classifier works with the group of tweets and does not take user's profile into account. This facilitates an analysis of user behavior and trends. The Big Five model enables the identification of personality traits through linguistic information. This can be considered as "multi-label classification" problem which is then transformed into five binary classification problems. A multilayer neural network classification algorithm is applied to determine personality traits. Further, the system makes use of the Hadoop framework to predict personality traits of multiple individuals at the same time. In twitter relevant information may not be available for analysis due to privacy issues. People may create fake accounts, or fake some information. This affects the results of personality prediction.

## LIMITATIONS

Relevant information not available for analysis due to privacy issues. People may create fake accounts, or fake some information. Tweets may be written in slang language and contain special characters. Information from Other sources such as linkedin, facebook etc can also be combined to provide better prediction.

# 10)EMOTION DETECTION FROM TEXT

**Shiv Naresh Shivhare and Prof. Saritha Khethawat**

Emotion Detection in text documents is essentially a content – based classification problem involving concepts from the domains of Natural Language Processing as well as Machine Learning. In this paper emotion recognition based on textual data and the techniques used in emotion detection are discussed. In this paper, methods which are currently being used to detect emotion from text are reviewed such as Keyword spotting technique, Lexical Affinity Method, Learning-based Methods, Hybrid Methods in step by step manner also its limitations and a new system architecture is proposed, which would perform efficiently. The proposed architecture contains two main components: Emotion Ontology, Emotion Detector. By the proposed algorithm we can find out the score of primary emotion classes. Emotion class with highest score will be decided as the final emotion class for the blog.

# 11)Modelling Empathy and Distress in Reaction to news Stories What the paper is doing

Well – written. Several news stories were selected and formed into a corpus and were evaluated by the text author themselves rather than having to approach a third party. Following tokenizing and rating data. The methodology follows

three models, one goes with Ridge regression, and the second comprises of a Feed

forward neural network and finally, a CNN.

**Criticism**

Paper could have gone into more details with respect to the methodology.

# 12)BIG FIVE PERSONALITY PREDICTION FROM SOCIAL MEDIA DATA USING MACHINE LEARNING TECHNIQUES

**Suman Maloji, Kasiprasad Mannepalli, Navya Sravani. J, K. Bhavya Sri, C. Sasidhar**

This paper aims to predict human personality by considering five features such as Openness, Agreeableness, Neuroticism, Extroversion, and Conscientiousness. thThe main aim is to address whether social media profiles can predict personality traits of a person. If so, various discoveries on the effects of personality and behavior variables can be incorporated into online user interactions as well as the use of social media profiles to help people understand each other better. They considered four machine learning algorithms namely SVM classification, Random Forest algorithm, Naïve Bayes Algorithm and Logistic regression to comparatively predict the user's personality accurately. To run this project they used a twitter dataset which contains tweets and user details. Among the machine learning algorithms used- SVM algorithm topped the accuracy report when compared with the other algorithms. Also, among the different tweets accessed from the database, users seem to present more conscientiousness with their routine lives when compared with the other four personality traits among the Big Five Personality traits prediction.

# PROPOSED METHODOLOGY

## EMOTION CLASSIFICATION

- **Data Preprocessing**

We are loading our textual data in a data frame. In order to make our textdata cleaner we need to perform some text preprocessing: removing punctuations ,removing stopwords, removing emails, HTML tags, website, and unnecessary links, removing contraction of words, normalisation or words

To make text-preprocessing easier a library named *text_hammer* is being used.After building text-preprocessing function we need call it on our dataframe. only training data need to be cleaned, not test and validation data.

- **Label encoding**

The sentiment category in our data frame needs to be converted into some numbers in order to pass into the model. Using a dictionary we are encoding our sentiment categories *{'joy':0,'anger':1,'disgust':2,'sadness':3,'fear':4,'surprise':5,'Neutral:6}*.

We have encoded our category by assigning them numbers now it's time to convert categories into categorical data.

- **Tokenization**

We convert our text corpus into some integer numbers.Tokenizer class converts a sentence into an array of numbers by assigning them numbers based on their frequency.Only the top "**num_words**" that is most frequent words will be taken into account. Only words known by the tokenizer will be taken into account hence we

have concatenated our train and test data to increase the vocabulary for the tokenizer.The method **fit_on_texts()** fits the text data to the tokenizer. It takes a list of sentences.Loading the pertained glove vector usingthe *gensim* library.

More dimension means more deep meaning of words but it may take a longer time to download.Now map the vocabulary learned by the tokenizer and create a weight matrix.**tokenizer.word_index.items()** returns a dictionary of unique words as key and frequency as value.

- ## Model

So far we preprocessed our data, converted our y_label into categorical data, mapped our vocabulary into the vector using word2vec.It's time to design our Bi-LSTM model. We already have created a word-embedding matrix. to feed our word_embedding matrix in our training we would use an embedding layer.There are 3 parameters in embedding layers.

- **input_dim** : Vocabulary Size( number of unique words for training)
- **output_dim** : Length of the vector for each word(embedding dimension)
- **input_length** : Maximum length of a sequence
- **trainable :** It's False, which means it will only use a given weight matrix
- **EMBEDDING_DIM = 100** means the embedding layer will create a vector in 100 dimensions.
- While Stacking RNN, the former RNN layers should be set **return_sequences** to True so that the following RNN layer layers can have the full sequence as input.
- **history_embedding** keeps the history of model training

## PERSONALITY PREDICTION

- **Data splitting and target variables**

With a split ratio of 80-20, it was decided that the features would be other attributes of the table barring the first one, which would be the target variable – Consciousness.

Following which there was a model made on the same attributes.

- **Model**

From a plethora of too many models, one such model we went with in order to predict the other attributes was a neural network regression model with ReLu activation functions as its inner layer and a softmax layer as the topmost layer. The model predicted the target variable successfully.

- **Gradient Boosting**

We also wanted to find out the performance metrics of the model if it operated on few gradient boosting models. Models such as XGBoost and LightGBM were invoked and their loss metrics were compared with the performance of the model during the regression. The test RMSE was slightly higher than the training RMSE which showed a lesser chance of error.

# APPENDIX

# Implementation / Code

# EMOTION CLASSIFICATION

### LOADING THE DATA

```
import numpy as np
import pandas as pd
```

```
[ ] import pandas as pd # importing the dataset
    df_train = pd.read_csv('/content/EC_PROJ_DATASET (1) (1).csv', encoding='utf-8')
    df_pred = pd.read_csv('/content/messages_test_features_ready_for_WS_2022 (1) (1).csv',encoding='utf-8')
```

### IMPORTING LIBRARIES

```
from sklearn.model_selection import train_test_split

train, test = train_test_split(df_train, test_size=0.2, random_state=25)
```

```
[ ] !pip install spacy
```

```
[ ] !pip install text_hammer
    import text_hammer as th
```

```
%%time
from tqdm._tqdm_notebook import tqdm_notebook
tqdm_notebook.pandas()
```

```
CPU times: user 448 µs, sys: 912 µs, total: 1.36 ms
Wall time: 1.24 ms
```

### DATA PREPROCESSING

```
[ ] def text_preprocessing(df,col_name):
        column = col_name
        df[column] = df[column].progress_apply(lambda x:str(x).lower())
        df[column] = df[column].progress_apply(lambda x: th.cont_exp(x))
        df[column] = df[column].progress_apply(lambda x: th.remove_emails(x))
        df[column] = df[column].progress_apply(lambda x: th.remove_html_tags(x))
        df[column] = df[column].progress_apply(lambda x: th.remove_special_chars(x))
        df[column] = df[column].progress_apply(lambda x: th.remove_accented_chars(x))
        df[column] = df[column].progress_apply(lambda x: th.make_base(x)) #ran -> run,
        return(df)
```

```
[ ] df_cleaned_train = text_preprocessing(train, 'essay')
```

```
100% 1488/1488 [00:00<00:00, 30983.62it/s]
100% 1488/1488 [00:06<00:00, 155.02it/s]
100% 1488/1488 [00:00<00:00, 5849.81it/s]
100% 1488/1488 [00:01<00:00, 1651.86it/s]
100% 1488/1488 [00:00<00:00, 4188.99it/s]
100% 1488/1488 [00:00<00:00, 7375.55it/s]
```

**LABEL ENCODING**

```
[ ] Bf_cIeaued_train['emtion'j = df_c1eaued_tralu.emotion.replace({'joy':0,'anger':1,'d1sgust':2,'saduess':3,'fear':4,'surprise':5,'neutral':6})
```

```
[ ] froa tensorfTcm.keras.utils import to_categorical
     y_traTn = to_categaricaT(df-_cleaned_train .ection. values)
```

TOKENIZATION

```
[ ] froc keras.preprocessing.text imaort Tokenizer
    nun words-tee0
    athis eeans 16800 unique uords can be taken
    tokenizer=Tokenizer(nua_uords, Lo¥ter=TrMe)
    df_iota1 - pd . concat([ df_cleaned_t rain[ ' essay ' ] test. essay ] , axis - 8)
    tokenizer . +it_on_text s(df-_tota I )
```

```
[ ] froa Keras *prepr oces sing. sequence import gad sequences
```

```
[ ] X_tra1n-token1zer.te"t"_ta_*e' luen' es df_c Jeaned_tra in[ ' essay ' ] ) # this converts texts inta some nur ric seouences
    M_traiu_pad-pad_sequences(M_train,ma*len-300,paddiug='post') 0 this makes the length of all numeric sequences equal
    M_test =tokenizer.texts_to_seauences(test.essay)
    M_test ad - pad_seouences(X_test,maxlen = 300, padding = 'post')
```

```
    import gens*m.down1oader as api
    g1ove ensia - api. load(' gLove-wiki - gigaword - 10B') t60 dimension
```

```
        |+ word ;n glove ensLn.wv . nacab:
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: DeprecationWarning: Call to deprecated `wv` (Attribute will be removed in 4.0.0, use self instead).
  This is separate from the ipykernel package so we can avoid doing imports until
```

**Bi-LSTM madel**

```
[ ] {rom teusorflow.keras.mde1s import Sequential
    from teusorflow.<eras.layers import Deuse, L51, Embeddiug,Bidirectiona1
    import tensor-flov
    froa \ensorl1afii.cozpaL.v1.keras.Layers import cuDNNLsTn
    froa tensorflou. <eras. layers inport Dropout
```

```
    c1ass_nun = 7
    «odeL - Sequent Tal( )
    -xleL. add{ fn bedding(input_d1m = nun_•ords,
     output d*m - EPI8E00IKG Dlk,
     input lengtfi- x trainp ad* shape[tJ,
     weights - [gens in weight natrixJ,trainabLe - ralse))
    mxleL. add(Dropout (e. 2))
    m›del. add( Bidi rectional(Cu0NNLSTN(186, return_sequences-True) ))
    m›deL. add(Dropout (0.2) )
    model.add(Bidirectional(CuDNNLSTM(200,return_sequences=True)))
    nxxiel.add(Dropout(8.2))
    nx›del. add(Midi rectional(Cu0iiNcSrH(l8B, return_sequences-ralse)) )
    nodel . add(oense(class_nus, activation = 'softinax ' ))
    n xiel . tonpile (loss = ' categori cal_cros sentropy ', optimizer = 'Adam ',metrics = ' ac turacy ' )
```

```
[ ]  history_embedding = model.fit(X_train_pad, y_train,
                                   epochs = 50, batch_size = 128,
                                   verbose = 1, validation_split=0.3)
```

```
Epoch 36/50
9/9 [==============================] - 2s 214ms/step - loss: 0.0395 - accuracy: 0.9846 - val_loss: 3.6487 - val_accuracy: 0.3736
Epoch 37/50
9/9 [==============================] - 2s 214ms/step - loss: 0.0195 - accuracy: 0.9962 - val_loss: 3.7075 - val_accuracy: 0.3826
Epoch 38/50
9/9 [==============================] - 2s 212ms/step - loss: 0.0265 - accuracy: 0.9914 - val_loss: 3.5639 - val_accuracy: 0.3714
Epoch 39/50
9/9 [==============================] - 2s 212ms/step - loss: 0.0304 - accuracy: 0.9914 - val_loss: 3.5482 - val_accuracy: 0.3758
Epoch 40/50
9/9 [==============================] - 2s 211ms/step - loss: 0.0317 - accuracy: 0.9894 - val_loss: 3.6596 - val_accuracy: 0.3826
Epoch 41/50
9/9 [==============================] - 2s 212ms/step - loss: 0.0716 - accuracy: 0.9731 - val_loss: 3.6627 - val_accuracy: 0.3669
Epoch 42/50
9/9 [==============================] - 2s 210ms/step - loss: 0.0490 - accuracy: 0.9846 - val_loss: 3.6254 - val_accuracy: 0.3915
Epoch 43/50
9/9 [==============================] - 2s 209ms/step - loss: 0.0386 - accuracy: 0.9885 - val_loss: 3.5354 - val_accuracy: 0.3893
Epoch 44/50
9/9 [==============================] - 2s 210ms/step - loss: 0.0262 - accuracy: 0.9933 - val_loss: 3.5069 - val_accuracy: 0.4027
Epoch 45/50
9/9 [==============================] - 2s 211ms/step - loss: 0.0221 - accuracy: 0.9952 - val_loss: 3.6099 - val_accuracy: 0.4116
Epoch 46/50
9/9 [==============================] - 2s 210ms/step - loss: 0.0247 - accuracy: 0.9914 - val_loss: 3.5484 - val_accuracy: 0.4206
Epoch 47/50
9/9 [==============================] - 2s 211ms/step - loss: 0.0139 - accuracy: 0.9981 - val_loss: 3.6632 - val_accuracy: 0.3915
Epoch 48/50
9/9 [==============================] - 2s 211ms/step - loss: 0.0111 - accuracy: 0.9971 - val_loss: 3.6424 - val_accuracy: 0.4072
Epoch 49/50
9/9 [==============================] - 2s 210ms/step - loss: 0.0223 - accuracy: 0.9962 - val_loss: 3.6244 - val_accuracy: 0.3982
Epoch 50/50
9/9 [==============================] - 2s 210ms/step - loss: 0.0094 - accuracy: 0.9981 - val_loss: 3.6366 - val_accuracy: 0.3915
```

# PERSONALITY PREDICTION:

**IMPORTING THE DATASET**

```
from google.colab import files
uploaded = files.upload()
import io
df = pd.read_csv(io.BytesIO(uploaded['personality.csv']),encoding = 'unicode escape')
df.head()
```

Choose Files No file chosen  Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving personality.csv to personality (2).csv

| | Conscientiousness | Openess | Extraversion | Agreeableness | Stability |
|---|---|---|---|---|---|
| 0 | 6.0 | 5.0 | 5.0 | 5.5 | 5.5 |
| 1 | 6.0 | 5.0 | 5.0 | 5.5 | 5.5 |
| 2 | 6.0 | 5.0 | 5.0 | 5.5 | 5.5 |
| 3 | 6.0 | 5.0 | 5.0 | 5.5 | 5.5 |
| 4 | 6.0 | 5.0 | 5.0 | 5.5 | 5.5 |

```
[ ] df.shape
```

```
(1860, 5)
```

**Checking the number of missing values**

```
[ ] df.isna().sum()
```

```
Conscientiousness    0
Openess              0
Extraversion         0
Agreeableness        0
Stability            0
dtype: int64
```

**Dropping null values**

```
[ ] df = df.dropna()
```

**Returning top 5 values of the dataframe**

```
[ ] x = df.iloc[:,1:]
    x.to_numpy()
```

```
array([[5. , 5. , 5.5, 5.5],
       [5. , 5. , 5.5, 5.5],
       [5. , 5. , 5.5, 5.5],
       ...,
       [7. , 7. , 7. , 7. ],
       [7. , 7. , 7. , 7. ],
       [7. , 7. , 7. , 7. ]])
```

**Returning top 5 values of the target variable**

```
[ ] y = np.array(df.iloc[:,0])
    y
```

```
array([6., 6., 6., ..., 7., 7., 7.])
```

**Splitting dataset into train and test sets**

```
[ ] from sklearn.model_selection import train_test_split

    x_train, x_test, y_train, y_test = train_test_split(x.to_numpy(),y,test_size = 0.2, random_state = 123)
```

```
[ ] VALIDATION_SPLIT=0.2
```

```
[ ] from sklearn.preprocessing import MinMaxScaler

    sc = MinMaxScaler()

    x_train = sc.fit_transform(x_train)
    x_test = sc.fit_transform(x_test)
```

Model building ant compiling

```
m>dg1 . add(oense{5B0, i nput_d inn = , activation = "re \u") )
rodel . add(oense{se, act i vat ion - "re\u“ ) )
eodel . add(eense{ ) )
rodel . sumary ( )
nodel : " sequenti at "
```

| Layer (type) | Output Shape | Param # |
|---|---|---|
| dense (oense) | (none, seo) | 2see |
| dense_z (ronse) | (None, z8o) | sete0 |
| dense_z (oense) | (None, se) | ses8 |

```
[ ]  model.compile(loss = "mean_squared_error", optimizer = 'adam', metrics = ['mae'])
     history = model.fit(x_train, y_train, batch_size = 32,epochs = 100,validation_split=VALIDATION_SPLIT)

     38/38 [==============================] - 0s 6ms/step - loss: 0.1865 - mae: 0.3117 - val_loss: 0.2828 - val_mae: 0.3513
     38/38 [==============================] - 0s 7ms/step - loss: 0.1765 - mae: 0.3022 - val_loss: 0.2676 - val_mae: 0.3555
     3S/3S [==============================]- Ba zms/step- loss: 8.&699- mme: B.193a- val_loss: S.£9S1- va1_mae: 0.387J
      #  B      ==:=:==::=:==:=====:::====] - 0s dms/step - loss: B.z6B0 - mae: B.Ze58 - val loss: z.z9a0 - val mae: 0.366a

     38/38 [==============================] - 0s 6ms/step - loss: 0.2197 - mae: 0.3413 - val_loss: 0.2827 - val_mae: 0.3735
     38/38 [==============================] - 0s 8ms/step - loss: 0.2495 - mae: 0.3678 - val_loss: 0.2841 - val_mae: 0.3558
     Epoch 10/100
     3B/38 [- --•---•- ••---•-•----•••----] - 8s 6ma/step - loss: 0.J676 - mae: 0.2879 - valloza: 0.I445 - val mae: B.3Z37
```

- History

```
[ ] score = model.evaluate(x test, y test,)
     print("\uloss:", score[0])
     pnint('nae', scone[1])

     12/t 2 [ =====--====-- -==========-=======] - 6s 2ns/step - loss: 6. 2252 - mae : 6.3 356

     mae 0.3156426954269409
```

```
# summarize history for loss
#p1t . pJot{history. history[ 'loss' ])
#pLt . pJot{history. history[ "nae ' ])
epLt . title( 'model loss' )
epLt .yJabeLo( ' loss ' )
up It . xlabeL( ' epoch' )
#p1t . legend{ [ 'train ' , ' test ' ], La= ' upper left ' )
epLt . shown)
```

```
[ ]  print(history.history.keys())
```

The curve of the loss is decreasing which means the model is good

- COMPARING WITH GRADIENT BOOSTING ALGORITHM

- Gradient Booming Regressor

```
[ §  + ron  s k learn . enseeb  e  import Grad :t ent8oos                sor
param -  'n_estinator s ' : 3, 'wax_dept h' : i,  '          rate ' : e.eea,  ‹ riterion ' :      'j
gbrmodel = GradientBoostingRegressor()

GradientBoostingRegressor()


The mean squared error (MSE) on test set: 0.6368
```

```
x traiu- xgreg.predict(x traiu)
print(np.sqrt(mean_squared_error(y_train,xg_train)))

2.133095373896218

x test = xgreg.predict(x_test)
print (np . sqct(mean squared error(y test,xy test )) )

2. 14443244891476
```

- LightGBM

```
import lightgbm as lg

lgmodel = lg.LGBMRegressor()

lgmodel.fit(x_train, y_train)
print(lgmodel)

LGBMRegressor()

[ ] lg_train = lgmodel.predict(x_train)
    print(np.sqrt(mean_squared_error(y_train,lg_train)))


[ ] lg_test = lgmodel.predict(x_test)
```

# CONCLUSION

## EMOTION CLASSIFICATION

Now let's go through a recap of what's being done:

We take the input text and then use tokenizer which converts into integer sequence. Use pad_sequence to make sequence length equal to the length used for training.Now pass the padded_sequence to model and call predict method, it will give us class index. Using the dictionary we defined earlier we changed the class index to the class label. Hence emotion classification has been implemented successfully.

## PERSONALITY PREDICTION.

This research shows the comparison of different feature extraction method along with different algorithm approach in building personality prediction system for multiple social media data sources. Future development of this experiment may utilize the use of larger training and testing dataset.
Furthermore, another comparison approaches such as implementing another pre-trained model such as ALBERT which is A Lite BERT for Self-supervised Learning of Language Representation, DistilBERT, and BigBird may also be a possible candidate to increase accuracy in the personality prediction system.

# REFERENCES


[1] Hardik, S &Gosai, Dhruvi& Gohil, Himangini. (2018). A REVIEW ON A EMOTION DETECTION AND RECOGNITION FROM TEXT USING NATURAL LANGUAGE PROCESSING.

[2] F. Calefato, F. Lanubile, N. Novielli. "EmoTxt: A Toolkit for Emotion Recognition from Text" In Proceedings of the Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos, {ACII} Workshops 2017, San Antonio, USA, Oct. 23-26, 2017, pp. 79-80.

[3] Zhou, Q., & Wu, H. (2018, October). NLP at IEST 2018: BiLSTM-attention and LSTM-attention via soft voting in emotion classification. In Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (pp. 189-194).

[4] Jabreel, M., & Moreno, A. (2019). A deep learning-based approach for multi-label emotion classification in tweets. Applied Sciences, 9(6), 1123.

[5] Credé, M., Harms, P., Niehorster, S., & Gaye-Valentine, A. (2012). An evaluation of the consequences of using short measures of the Big Five personality traits. Journal of personality and social psychology, 102(4), 874.

[6] Big Five Personality Detection Using Deep Convolutional Neural Networks. WaielTinwala, ShristiRauniyar. Dept. of Computer Science and Engineering, Delhi Technological University, Delhi, India

[7] Bai, S., Hao, B., Li, A., Yuan, S., Gao, R., & Zhu, T. (2013, November). Predicting big five personality traits of microblog users. In 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) (Vol. 1, pp. 501-508). IEEE. [8] Ambar,S., Jan,S., & Khan, F. (2019). Human emotion detection and classification in text mining. Pakistan Journal of Science,71(4),235.

[9] Kalghatgi, M.P., Ramannavar, M.M., &Sidnal, D.N. (2015). A Neural Network Approach to Personality Prediction based on the Big-Five Model.

[10] Shivhare, Shiv Naresh &Khethawat, Saritha. (2012). Emotion Detection from Text. Computer Science & Information Technology. 2. 10.5121/csit.2012.2237.

[11] Buechel, Sven &Buffone, Anneke &Slaff, Barry & Ungar, Lyle &Sedoc, João. (2018). Modeling Empathy and Distress in Reaction to News Stories. 4758-4765. 10.18653/v1/D18-1507.

[12] Suman Maloji, KasiprasadMannepalli, NavyaSravani. J, K. Bhavya Sri, C. Sasidhar.Volume-9 Issue-4, April 2020. Big Five Personality Prediction from Social Media Data using Machine Learning Techniques