NLP - COM3029 & COMM061

INDIVIDUAL COURSEWORK

Name: Anitha Kugur Parameshwarappa

URN: 6784786 (ak02923@surrey.ac.uk)

Group Number: 11

ABSTRACT

One of the most popular functions that have been used for a range of situations is text categorization. Text classification is most often used to group documents into categories such as news articles about business, sports, politics, or other events. It is also frequently used to recognize spam in emails, understand the tone of messages on social media sites like Twitter, or rate products on online stores like Amazon. According to how we want our application to behave, the input text may finally be assigned just one label (multi-class) or several labels (multi-label) during the prediction phase.

INTRODUCTION

The objective is to create a multi-class classifier prototype for a data topic. In this case, the data topic is the GoEmotions dataset, which consists of 58k carefully selected human-annotated Reddit comments that fall into one of 27 distinct mood categories or are "neutral." The following list of feelings are categorized as emotions: admiration, amusement, anger, annoyance, caring, love, approval, curiosity, disapproval, confusion, desire, disgust, disappointment, excitement, embarrassment, joy, gratitude, fear, pride, optimism, relief, realization, remorse, surprise ,sadness &neutral.

GROUP DECLARATION:

In this study, we have chosen to sentiment using a 14-label classification system, which allows for a more nuanced and detailed understanding of the emotions expressed in the text. The 14 labels selected for this project are:

1. Neutral 2. Surprise 3. Love 4. Fear

5. Joy 6. Gratitude 7. Approval 8. Disapproval

9. Confusion 10. Sadness 11. Desire 12. Optimism

13. Realization 14. Pride

This multi-class classification approach enables a more comprehensive sentiment recognition by incorporating diverse emotions. The chosen labels cover a wide spectrum of emotions, ranging from positive (e.g., love, joy, gratitude) to negative (e.g., fear, sadness, disapproval), as well as those that are more complex or context-dependent (e.g., surprise, confusion, realization).

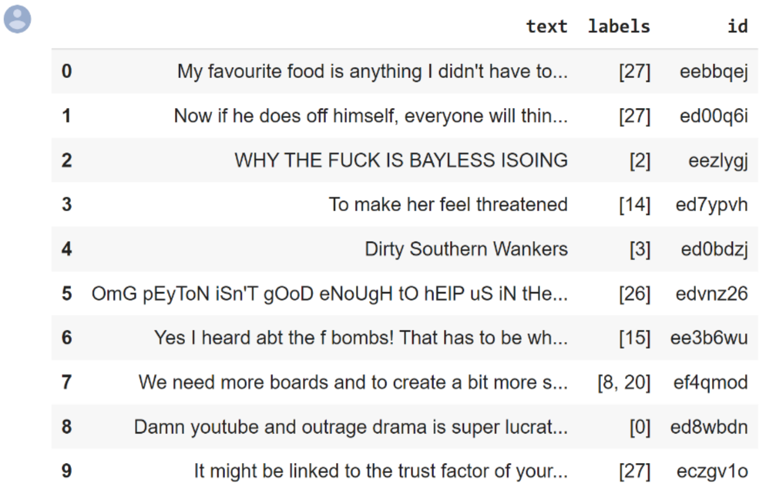
Then, it highlights the necessary steps for planning experiments, including defining the scope, data source, sample size, tools, and techniques, sentiment recognition, and conclusion of the experiments. The text also mentions the use of GitHub, Google colab, and Dataset as common development environments. Lastly, the individual tasks are separated into data set preparations, algorithms, pre-trained models, setting up hyperparameters and experimental variations, and tracking progress.

This customized 14-label sentiment recognition will facilitate a deeper understanding of the underlying emotions present in text data, allowing for more informed decision-making and enhanced communication in various contexts, such as marketing, customer service, product development, and social media monitoring.

DATA IMPORTING

The simplified version of the go_emotions dataset has been imported and the train, test, and validation datasets are being loaded onto respective data frames using the pandas library.

Each data frame has three columns, namely text, labels, id.



	text	labels	id
0	My favourite food is anything I didn't have to...	[27]	eebbqej
1	Now if he does off himself, everyone will thin...	[27]	ed00q6i
2	WHY THE FUCK IS BAYLESS ISOING	[2]	eezlygj
3	To make her feel threatened	[14]	ed7ypvh
4	Dirty Southern Wankers	[3]	ed0bdzj
5	OmG pEyToN iSn'T gOoD eNoUgH tO hElP uS iN tHe...	[26]	edvnz26
6	Yes I heard abt the f bombs! That has to be wh...	[15]	ee3b6wu
7	We need more boards and to create a bit more s...	[8, 20]	ef4qmod
8	Damn youtube and outrage drama is super lucrat...	[0]	ed8wbdn
9	It might be linked to the trust factor of your...	[27]	eczgv1o

Fig.,(1) The go_emotions dataset

DATA ANALYSIS & VISUALISATION

Data Visualisation is a way to represent the data in graphical or visual format. It helps get insights from the dataset. The labels range between 0-27 with each label assigned to an emotion. Data visualisation using several plots like word cloud, bar plot, pie-chart are shown the figures below.

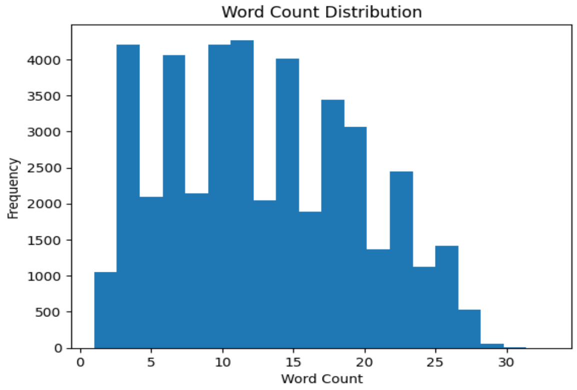


Fig.,(2)Word-count distribution

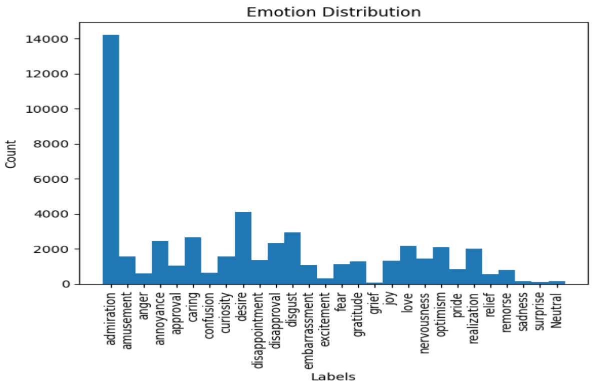


Fig.,(3) Emotion distribution

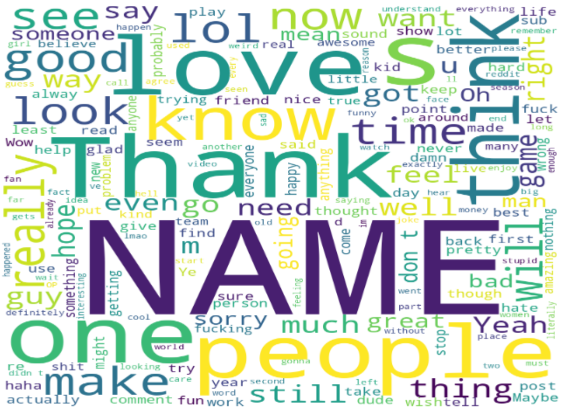


Fig.,(4) Word cloud

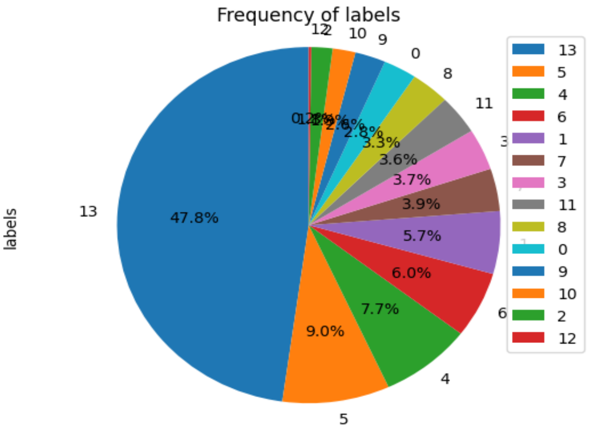


Fig.,(5) Frequency of labels

- Fig.,(1) shows the distribution of word count.
- Fig.,(2) depicts the emotion distribution, i.e., how many times each emotion has occurred in the dataset.
- Fig.,(3) shows the word cloud plot of the most frequently occurring words in the text data.
- Fig.,(4) is the picture of the pie-chart distribution of emotion labels after data pre-processing and it shows the occurrence of our selected labels (Surprise, Love, Fear, Joy, Gratitude, Approval, Disapproval, Confusion, Sadness, Realise, Desire, Optimism, Pride, Neutral) in percentage.
- We could see that neutral is the most commonly occurred label out of our selected labels.

EXPERIMENT 1 : Data PreProcessing

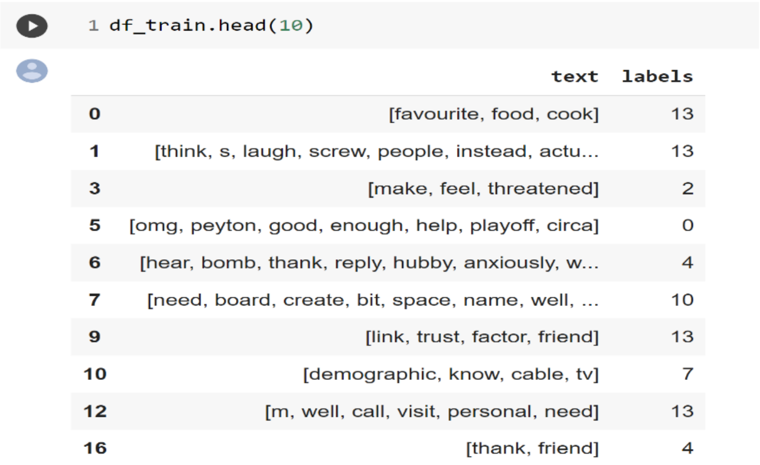
Data pre-processing is a critical step in the data analysis process and can greatly affect the accuracy and reliability of the final analysis results**.**

The following below is the pipeline that has been followed to pre-process the data.

Pre-processing technique 1:

a. Drop the unwanted columns (id-column from the data frame)
b. Drop the unwanted labels and keep only selected 14 labels (labels other than our selected labels)
c. Eliminate special characters, and numbers using regular expressions.
d. Removal of emoticons from the text column.
e. Remove English stop words using the nltk library.
f. Covert the texts to lowercase
g. Perform lemmatization to get the root words from nouns, verbs, adjectives, and adverbs.
h. Perform tokenization using spacy.
i. After reducing the labels to 14, map the labels from 0-27 to 0-14 using python dictionary.
j. Handling multiple labels by taking a single label out of the list into consideration.
k. Transform the list of labels to integer values.
l. Some neutral samples have been removed from the dataset since there was a huge amount of neutral label occurrence as compared to the other labels.

After pre-processing the dataset looks as shown in the figure below,



```
df_train.head(10)
```

	text	labels
0	[favourite, food, cook]	13
1	[think, s, laugh, screw, people, instead, actu...	13
3	[make, feel, threatened]	2
5	[omg, peyton, good, enough, help, playoff, circa]	0
6	[hear, bomb, thank, reply, hubby, anxiously, w...	4
7	[need, board, create, bit, space, name, well, ...	10
9	[link, trust, factor, friend]	13
10	[demographic, know, cable, tv]	7
12	[m, well, call, visit, personal, need]	13
16	[thank, friend]	4

Fig.,(6) Data after pre-processing

In the pre-processing technique 1 the variation being done in the data pre-processing are,

pre-processing technique 2:

a. Perform stemming using nltk PorterStemmer instead of lemmatization.
b. This time stop words have not been eliminated.
c. Except this all the other steps are the same as above.

EXPERIMENT 2: NLP Algorithms/Techniques

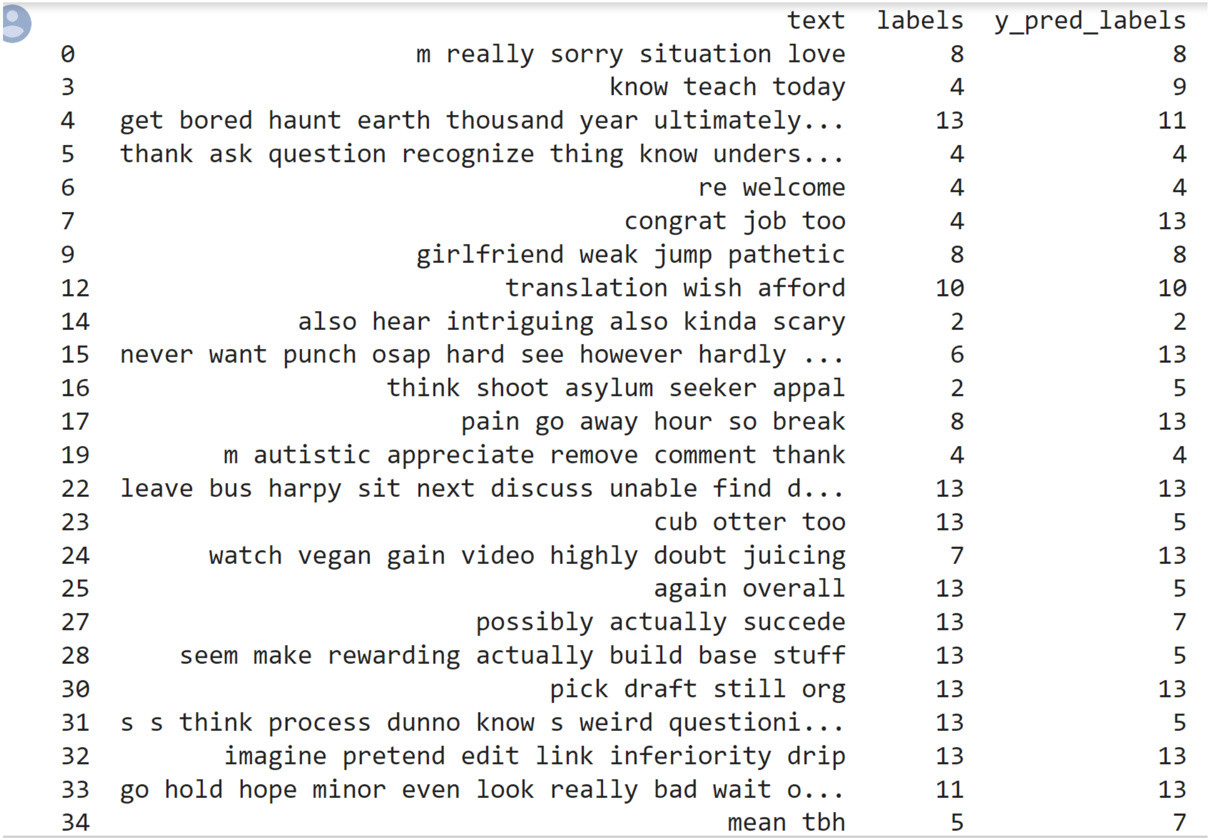
Experimented with two algorithms namely, SVM and LSTM

1. Support Vector Machine Classifier (SVM):

- SVM has been widely used in text classification and image classification. SVM has been known for its versatility, effectiveness, and ability to handle high-dimensional data.
- The word vectors of training data and validation data have been converted into numpy arrays so as the label values. And they are being fed to the model. The SVM kernel being used in the SVM model is ‘linear’.
- After feeding the dataset with pre-processing technique 1, the SVM model resulted with training accuracy equal to 49% and validation accuracy equal to 46%.

- Fig.,(5) shows the true labels and predicted labels by the SVM model. The text and true labels are taken from the test dataset.
- Fig.,(6) shows the confusion matrix, which summarizes actual labels against the predicted labels generated labels by the model.

We could see that the model has predicted some labels correctly like 8,4,13,10,2 fewer times it has also wrongly predicted those labels. The model has wrongly predicted the labels 7,11,5.



```
                                                 text  labels  y_pred_labels
0                       m really sorry situation love       8              8
3                                    know teach today       4              9
4   get bored haunt earth thousand year ultimately...      13             11
5   thank ask question recognize thing know unders...       4              4
6                                          re welcome       4              4
7                                     congrat job too       4             13
9                       girlfriend weak jump pathetic       8              8
12                            translation wish afford      10             10
14              also hear intriguing also kinda scary       2              2
15  never want punch osap hard see however hardly ...       6             13
16                    think shoot asylum seeker appal       2              5
17                         pain go away hour so break       8             13
19         m autistic appreciate remove comment thank       4              4
22  leave bus harpy sit next discuss unable find d...      13             13
23                                      cub otter too      13              5
24        watch vegan gain video highly doubt juicing       7             13
25                                      again overall      13              5
27                          possibly actually succede      13              7
28      seem make rewarding actually build base stuff      13              5
30                               pick draft still org      13             13
31  s s think process dunno know s weird questioni...      13              5
32         imagine pretend edit link inferiority drip      13             13
33  go hold hope minor even look really bad wait o...      11             13
34                                           mean tbh       5              7
```

Fig.,(5) True labels and predicted labels

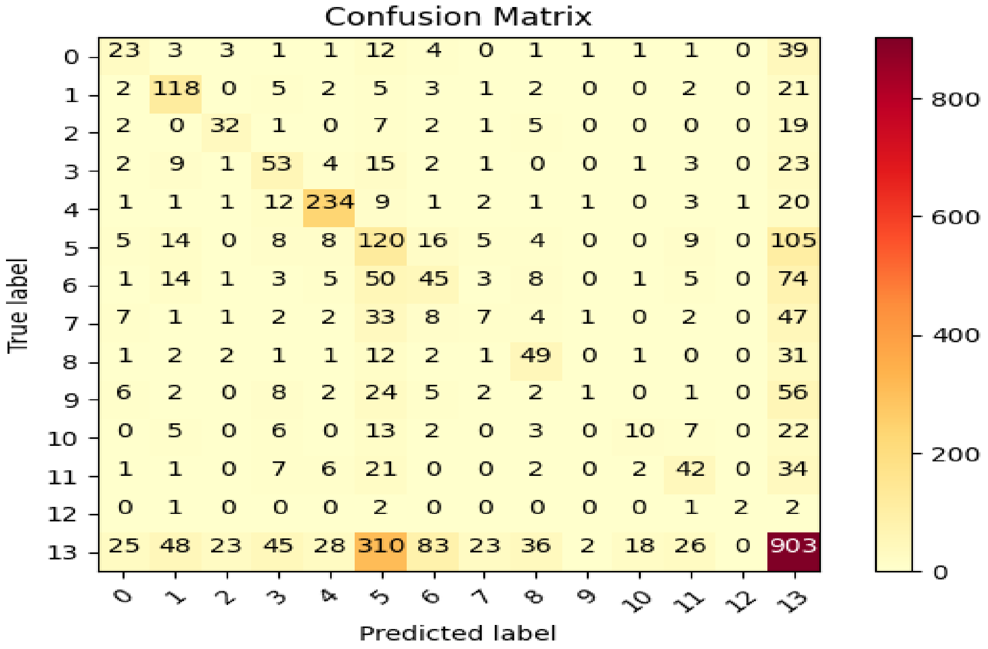


Fig.,(6) Confusion matrix of SVM model being trained using pre-processing technique 1

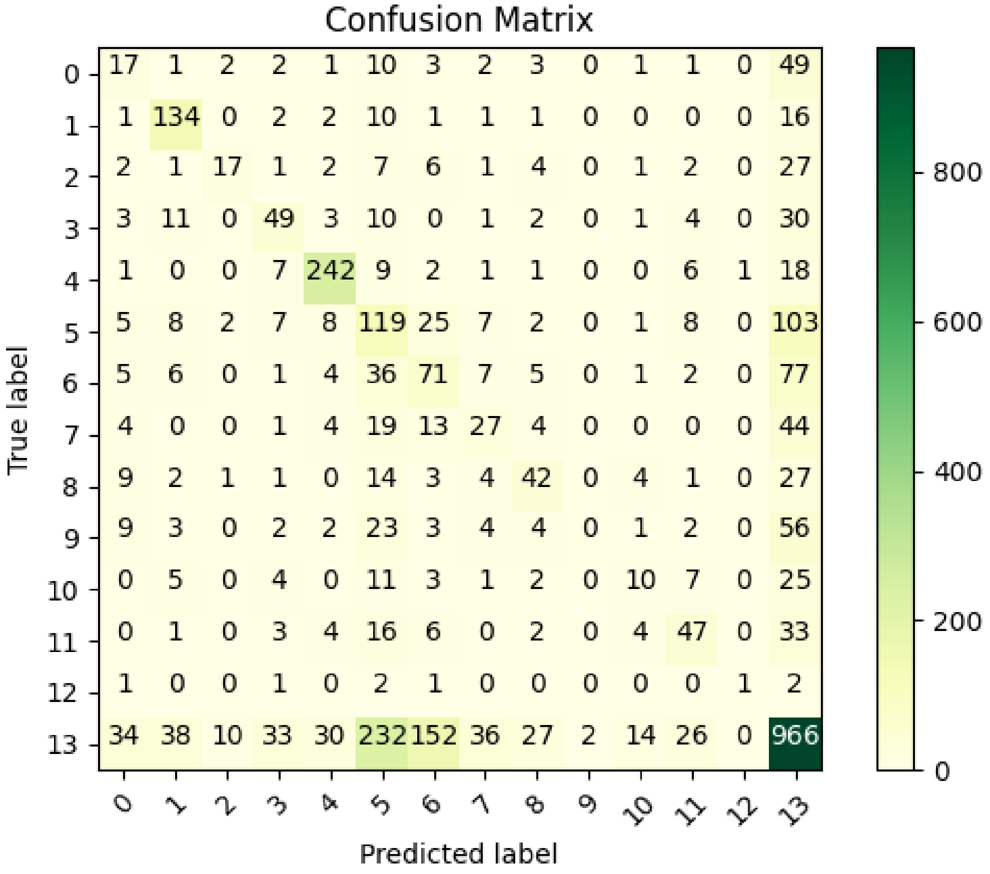


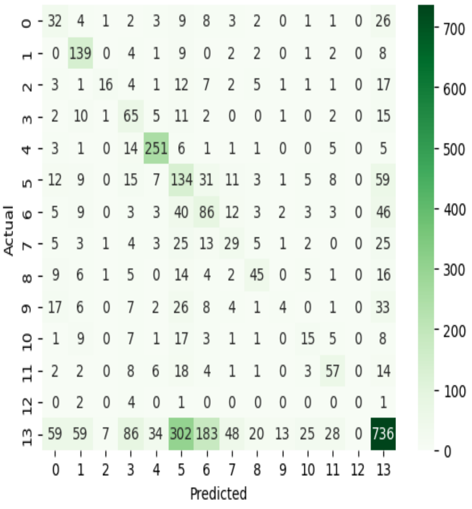
Fig.,(7) Confusion matrix of SVM model being trained using pre-processing technique 2

OBSERVATION

The SVM model accuracy improved slightly after using the pre-processing technique 2. The validation accuracy reached 47.5% & Test accuracy improved to 51.7%.

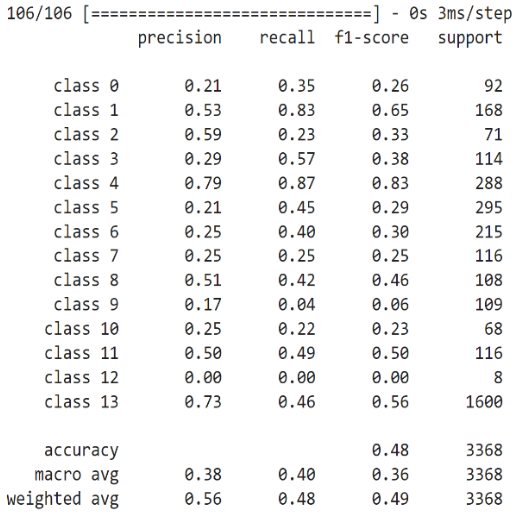
2. <u>Long Short Term Memory (LSTM):</u>

- Long short term memory is a kind of Recurrent neural network.
- The first LSTM layer has 64 units. The input data has a feature vector of size 300 and the second layer is another LSTM layer with 32 units.
- After the LSTM layers, a Dropout layer is added with a dropout rate of 0.2. The Dropout layer helps to prevent overfitting by randomly dropping out 20% of the units during training.
- Finally, a Dense layer with 14 units and ‘softmax’ is used as activation function to produce the final output.
- The model compilation is done using the ‘Sparse_categorical_crossentropy’ as a loss function, the 'Adam' as an optimization algorithm, and ‘accuracy’ being set as an evaluation metric.
- The embedding technique being used here is Glove vector embedding and data pre-processing technique 2 is being used. The LSTM model is trained for 20 epochs.
- The obtained accuracy after training the model with the pre-processed dataset, **Training accuracy= 52%, Test accuracy= 45%, and Validation accuracy= 48%.**



	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	32	4	1	2	3	9	8	3	2	0	1	1	0	26
1	0	139	0	4	1	9	0	2	2	0	1	2	0	8
2	3	1	16	4	1	12	7	2	5	1	1	1	0	17
3	2	10	1	65	5	11	2	0	0	1	0	2	0	15
4	3	1	0	14	251	6	1	1	1	0	0	5	0	5
5	12	9	0	15	7	134	31	11	3	1	5	8	0	59
6	5	9	0	3	3	40	86	12	3	2	3	3	0	46
7	5	3	1	4	3	25	13	29	5	1	2	0	0	25
8	9	6	1	5	0	14	4	2	45	0	5	1	0	16
9	17	6	0	7	2	26	8	4	1	4	0	1	0	33
10	1	9	0	7	1	17	3	1	1	0	15	5	0	8
11	2	2	0	8	6	18	4	1	1	0	3	57	0	14
12	0	2	0	4	0	1	0	0	0	0	0	0	0	1
13	59	59	7	86	34	302	183	48	20	13	25	28	0	736

Fig.,(8) Confusion matrix for LSTM



```
106/106 [==============================] - 0s 3ms/step
              precision    recall  f1-score   support

     class 0       0.21      0.35      0.26        92
     class 1       0.53      0.83      0.65       168
     class 2       0.59      0.23      0.33        71
     class 3       0.29      0.57      0.38       114
     class 4       0.79      0.87      0.83       288
     class 5       0.21      0.45      0.29       295
     class 6       0.25      0.40      0.30       215
     class 7       0.25      0.25      0.25       116
     class 8       0.51      0.42      0.46       108
     class 9       0.17      0.04      0.06       109
    class 10       0.25      0.22      0.23        68
    class 11       0.50      0.49      0.50       116
    class 12       0.00      0.00      0.00         8
    class 13       0.73      0.46      0.56      1600

    accuracy                           0.48      3368
   macro avg       0.38      0.40      0.36      3368
weighted avg       0.56      0.48      0.49      3368
```

Fig.,(9) Classification Report of LSTM

EXPERIMENT 3: Text Featurisation/ Word Embedding Techniques

Word embedding in NLP is a technique to represent the textual data/ words into numerical format as vectors.

Experimentation has been done using two different pre-trained vector embeddings to convert the tokenized words into vectors. Pre-trained word embedding techniques offer improved performance and reduce computational time and cost.

a. GloVe word embedding

Glove embedding is a popular word embedding technique to represent words as numerical vectors in NLP. ‘**glove-wiki-gigaword-300’** has been used where, 300 represents the dimensionality of the vectors for each word.

b. Word2Vec word embedding

Word2Vec is another popular vector embedding technique developed by Google. ‘**word2vec-google-news-300’** from the gensim library which also provides 300 dimensionality of vectors for each word.

The accuracy of LSTM with word2Vec embedding being used to get the vectors of the tokenised words,

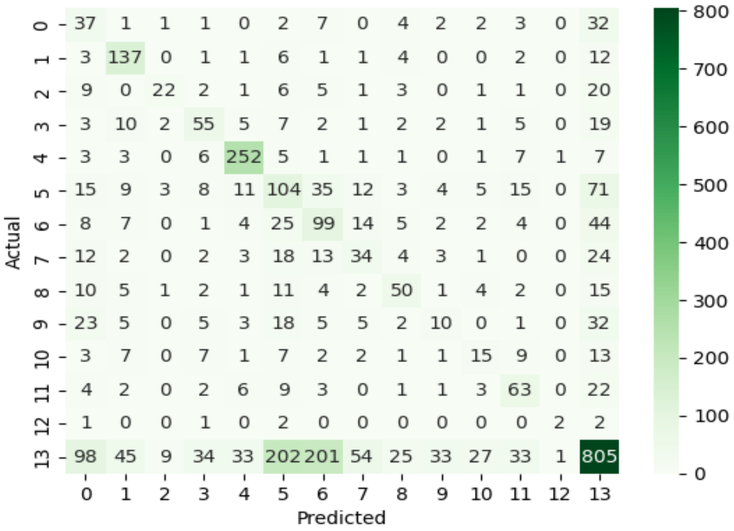
- Training accuracy= 54%
- Validation accuracy= 50%
- Test accuracy=51%

EXPERIMENT 4: Hyperparameters Tuning

Changes have been done on LSTM model by using Bidirectional LSTM with two bidirectional LSTM layers. First bidirectional layer with 64 inputs and an input length of 300. Second layer with 32 inputs. A third drop-out layer with a drop-out rate of 20% and finally a dense layer with 14 units and a softmax activation function. Hyperparameters like batch size, and epochs have been tuned, and also implemented early stopping, a regularization technique that monitors the validation loss and stops the training if validation loss does improve after ‘n’ consecutive epochs to prevent overfitting of the model and restores the model weights from epoch with lowest validation loss.

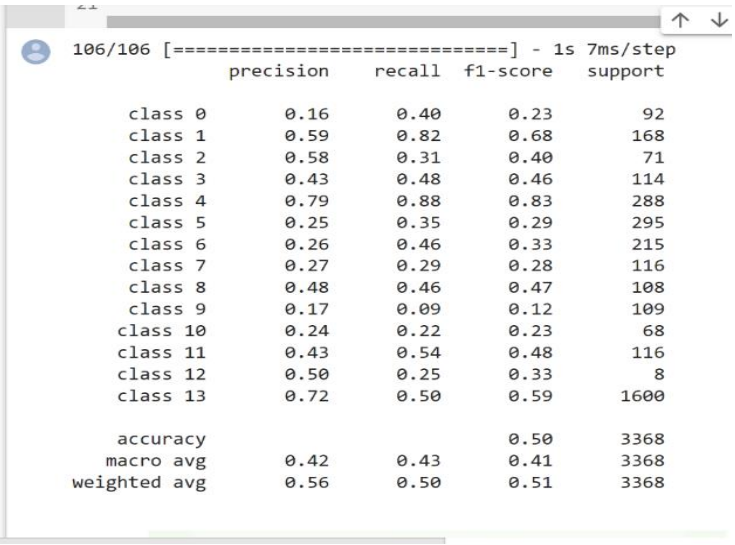
The accuracy obtained is as given below,

- Training accuracy= 54.4%,
- Test accuracy = 51.6%
- Validation accuracy= 51.6%



Actual \ Predicted	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	37	1	1	1	0	2	7	0	4	2	2	3	0	32
1	3	137	0	1	1	6	1	1	4	0	0	2	0	12
2	9	0	22	2	1	6	5	1	3	0	1	1	0	20
3	3	10	2	55	5	7	2	1	2	2	1	5	0	19
4	3	3	0	6	252	5	1	1	1	0	1	7	1	7
5	15	9	3	8	11	104	35	12	3	4	5	15	0	71
6	8	7	0	1	4	25	99	14	5	2	2	4	0	44
7	12	2	0	2	3	18	13	34	4	3	1	0	0	24
8	10	5	1	2	1	11	4	2	50	1	4	2	0	15
9	23	5	0	5	3	18	5	5	2	10	0	1	0	32
10	3	7	0	7	1	7	2	2	1	1	15	9	0	13
11	4	2	0	2	6	9	3	0	1	1	3	63	0	22
12	1	0	0	1	0	2	0	0	0	0	0	0	2	2
13	98	45	9	34	33	202	201	54	25	33	27	33	1	805

Fig,.(6) Confusion matrix for Bi-directional LSTM



```
106/106 [==============================] - 1s 7ms/step
              precision    recall  f1-score   support

     class 0       0.16      0.40      0.23        92
     class 1       0.59      0.82      0.68       168
     class 2       0.58      0.31      0.40        71
     class 3       0.43      0.48      0.46       114
     class 4       0.79      0.88      0.83       288
     class 5       0.25      0.35      0.29       295
     class 6       0.26      0.46      0.33       215
     class 7       0.27      0.29      0.28       116
     class 8       0.48      0.46      0.47       108
     class 9       0.17      0.09      0.12       109
    class 10       0.24      0.22      0.23        68
    class 11       0.43      0.54      0.48       116
    class 12       0.50      0.25      0.33         8
    class 13       0.72      0.50      0.59      1600

    accuracy                           0.50      3368
   macro avg       0.42      0.43      0.41      3368
weighted avg       0.56      0.50      0.51      3368
```

Fig.,(7) Classification report for Bi-directional LSTM

OBSERVATION

- Experimented with lower batch size of 32 and learning cycle of 60 epochs, observed that the lower batch size induces overfitting of the model.
- Experimented with dropout rate of 10% and optimizer as stochastic gradient descent (SGD) resulted in poor validation accuracy and training accuracy.

CONCLUSION

The best model as per the trials is the Bi-directional LSTM model with the accuracy obtained as below,

- Training accuracy= 55.4%
- Test accuracy = 53.2 %
- Validation accuracy= 53.2%

REFERENCES

[1] Joachims, Thorsten. "Text categorization with support vector machines: Learning with many relevant features." Machine Learning: ECML-98: 10th European Conference on Machine Learning Chemnitz, Germany, April 21–23, 1998 Proceedings. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.

[2] Wang, Jenq-Haur, et al. "An LSTM approach to short text sentiment classification with word embeddings." Proceedings of the 30th conference on computational linguistics and speech processing (ROCLING 2018). 2018.