NLU Programming Assignment - 1

(Report)

Links of different file :

Task 1,2,3 (Task_1,2,3)
Task 4(a) (Task4(a))
Task4(b),4(c) (Task4(b),(c))
Task4(d) (Task4(d))
Task4(f) (Task4(f))

Dataset : My team selected the dataset of the Coronavirus tweets NLP - Text Classification .On based on this dataset we did our task in the given assignment. My team has two members :

1) Aditya Mishra (M20MA201)
2) Atul Kumar Yadav (M20MA209)

Tasks :

1) **Define your own train-val-test split.**

   As we downloaded the train and test dataset separately from the given link. So we have to combine it and make a whole data set to create our own test-train-val split for the further processing.

   Following are the top four dataset details :

index	UserName	ScreenName	Location	TweetAt	OriginalTweet	Sentiment
0	3799	48751	London	16-03-2020	@MeNyrbie @Phil_Gahan @Chrisitv https://t.co/iFz9FAn2Pa and https://t.co/xX6ghGFzCC and https://t.co/I2NlzdxNo8	Neutral
1	3800	48752	UK	16-03-2020	advice Talk to your neighbours family to exchange phone numbers create contact list with phone numbers of neighbours schools employer chemist GP set up online shopping accounts if poss adequate supplies of regular meds but not over order	Positive
2	3801	48753	Vagabonds	16-03-2020	Coronavirus Australia: Woolworths to give elderly, disabled dedicated shopping hours amid COVID-19 outbreak https://t.co/bInCA9Vp8P	Positive
3	3802	48754	NaN	16-03-2020	My food stock is not the only one which is empty... PLEASE, don't panic, THERE WILL BE ENOUGH FOOD FOR EVERYONE if you do not take more than you need. Stay calm, stay safe. #COVID19france #COVID_19 #COVID19 #coronavirus #confinement #Confinementotal #ConfinementGeneral https://t.co/zrlG0Z520j	Positive
4	3803	48755	NaN	16-03-2020	Me, ready to go at supermarket during the #COVID19 outbreak. Not because I'm paranoid, but because my food stock is litteraly empty. The #coronavirus is a serious thing, but please, don't panic. It causes shortage... #CoronavirusFrance #restezchezvous #StayAtHome #confinement https://t.co/usmuaLq72n	Extremely Negative

For the split of the dataset , we splitted the whole dataset into Training (75%) and testing (25%) and then this training dataset of 75% again splitted into two parts : Training dataset(75% of Previously splitted training dataset) and Validation dataset (25% of Previously splitted training dataset). Hence we finally get the split of : **Training part (56.25%)**

Testing part (25%)

Validation part (18.75%)

2) text preprocessing pipeline .

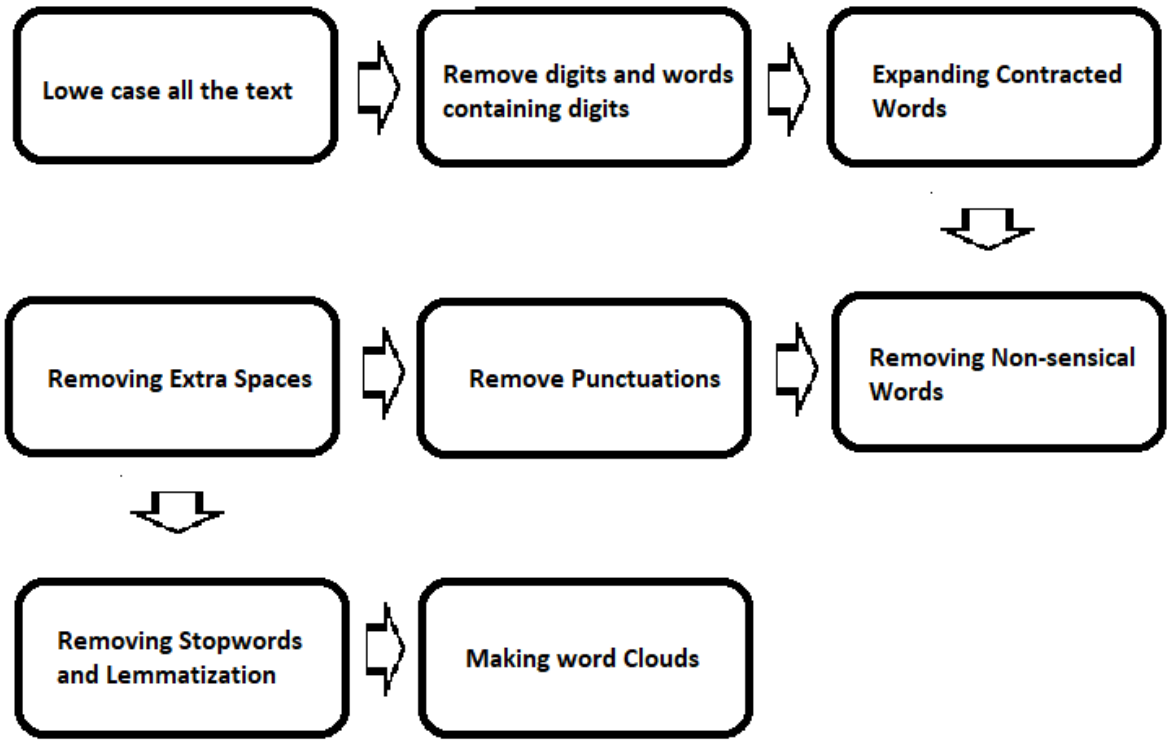


Fig:1. Text processing pipeline

We draw a word Cloud of the this provided dataset ,

Which is given below on the behalf of the words that came in the twitter text dataset.

Fig. 2 : Word Cloud

3) Developing ML Methods :

a) Model a **Naive Bayes classifier**.

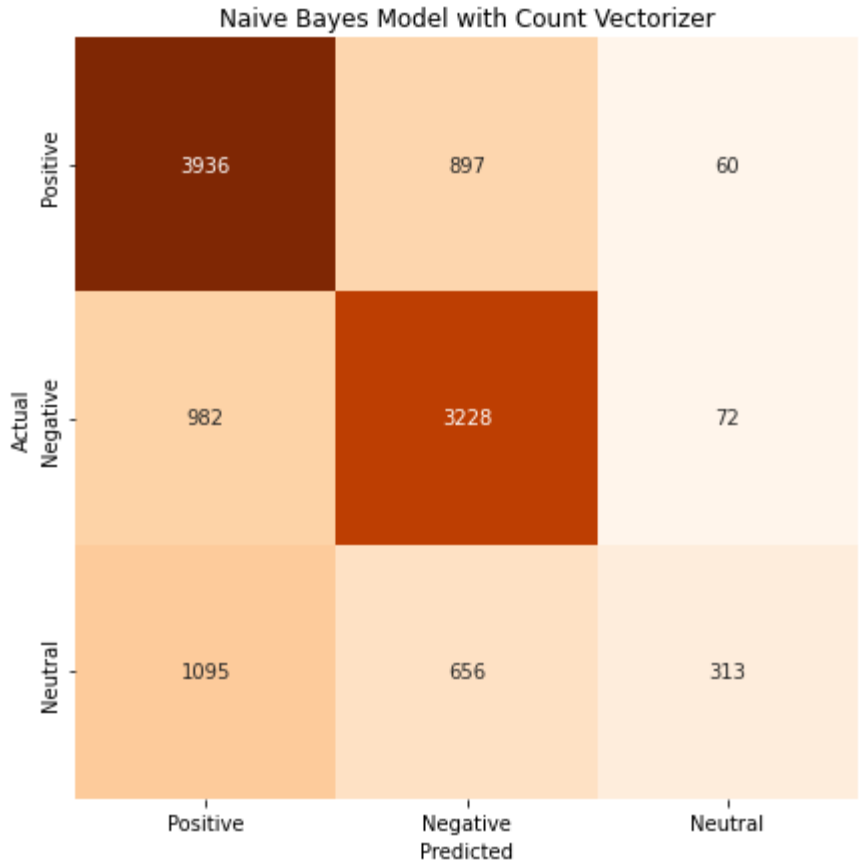
(i) **Count vectorizer features :**

We created a multinomial Naive Bayes Classifier using the Libraries after applying the Count vectorization technique and got the following Results :

Training Validation Accuracy : **66.070 %**

Testing Accuracy : 66.527 %

And we get the Confusion Matrix and their different Results below:



Classification Report

	precision	recall	f1-score	support
Positive	0.65	0.80	0.72	4893
Negative	0.68	0.75	0.71	4282
Neutral	0.70	0.15	0.25	2064
accuracy			0.67	11239
macro avg	0.68	0.57	0.56	11239
weighted avg	0.67	0.67	0.63	11239

(iI) **TF-IDF Features :**

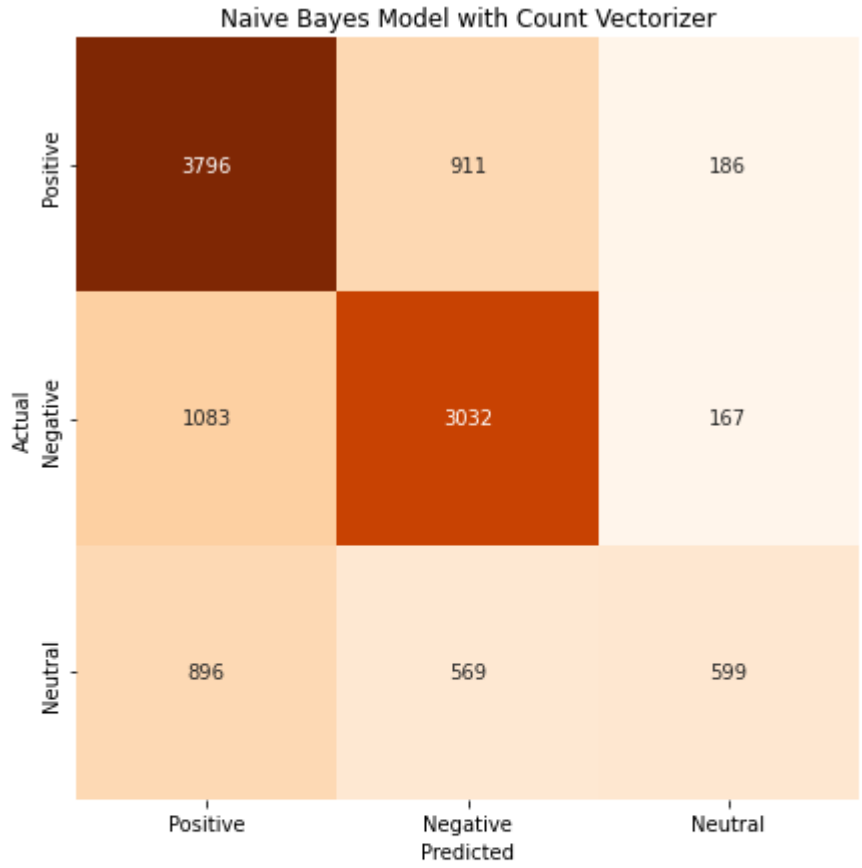
We created a multinomial Naive Bayes Classifier using the Libraries after applying the TF-IDF features technique and got the following Results :

Training Validation Accuracy : **64.94 %**

Testing Accuracy : 66.08 %

(Here we use the **Hyperparameter** as the Alpha as 0.1 and after we get the `AUC score of the model is 0.693868787655605` in the training part. And in the testing part `AUC score of the model is 0.7018393679699443.`)

And we get the Confusion Matrix and their different Results below:



Classification Report

	precision	recall	f1-score	support
Positive	0.66	0.78	0.71	4893
Negative	0.67	0.71	0.69	4282
Neutral	0.63	0.29	0.40	2064
accuracy			0.66	11239
macro avg	0.65	0.59	0.60	11239
weighted avg	0.66	0.66	0.65	11239

b) Model a **Decision Tree** with TF-IDF features.

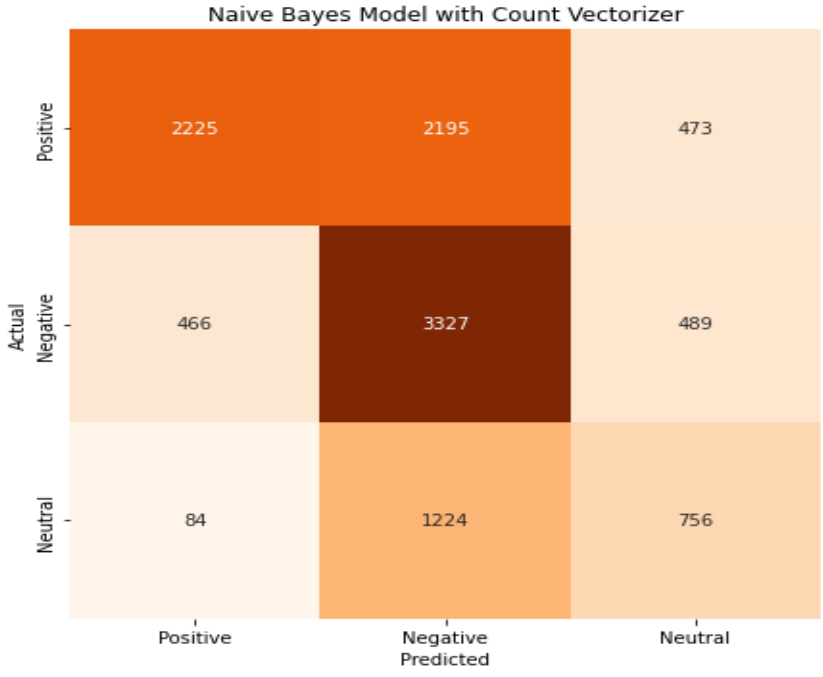
We train the Decision tree with the different tree's depth as Hyperparameter **depths are as [**`3,6,9,12,15,20`**]** and get the highest accuracy in the depth of size = **20**.

The following results are given below :

Training Validation Accuracy : **56.59%**

Testing Accuracy : 56.12 %

And we get the Confusion Matrix and their different Results below:



Classification Report

	precision	recall	f1-score	support
Positive	0.80	0.45	0.58	4893
Negative	0.49	0.78	0.60	4282
Neutral	0.44	0.37	0.40	2064
accuracy			0.56	11239
macro avg	0.58	0.53	0.53	11239
weighted avg	0.62	0.56	0.56	11239

4) Developing Deep Neural Networks :

(a) Model a **RNN** :

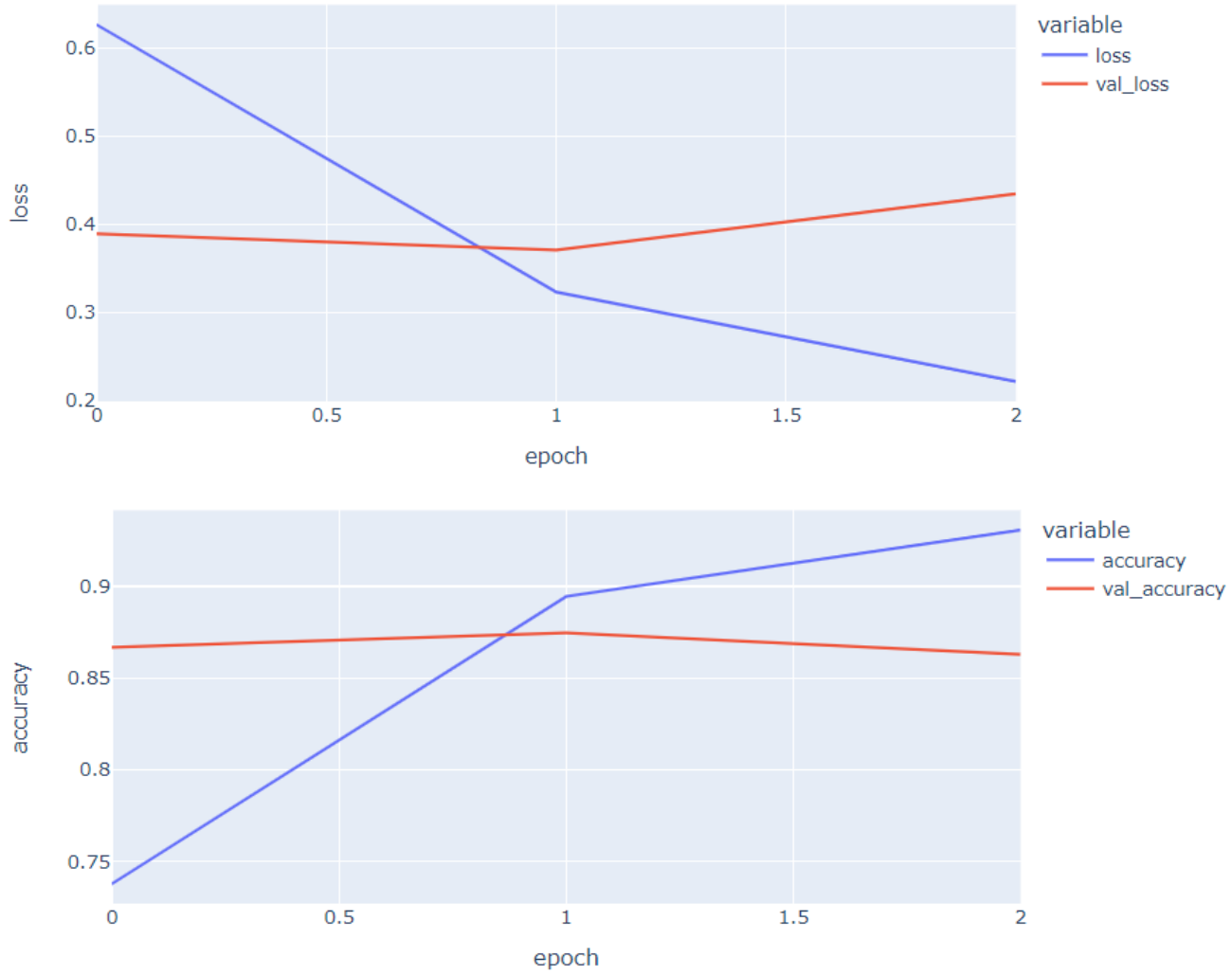
(i) 64 Hidden Vector Dimension :

We used the different details for training the RNN model , and these are as following : Embedding Dimension = 16
Embedding Type = Word Embedding
Pooling Layer as : MaxPool
Activation Function : “Relu”
2 Dropout Layers
“Softmax” function in the output Layer
Optimizer = “ADAM”
Loss = “Categorical CrossEntropy”

Batch size = 32 and No. of Epochs = 3 and Verbose = 2.

The **final training accuracy** is = **85.81 %**
final testing accuracy is = **84.76 %**

And the final graphs for the training accuracy vs epoch and loss vs epoch is given below:



(i) **256 Hidden Vector Dimension :**

We used the different details for training the RNN model , and these are as following : Embedding Dimension = 16

Embedding Type = Word Embedding

Pooling Layer as : MaxPool

Activation Function : “Relu”

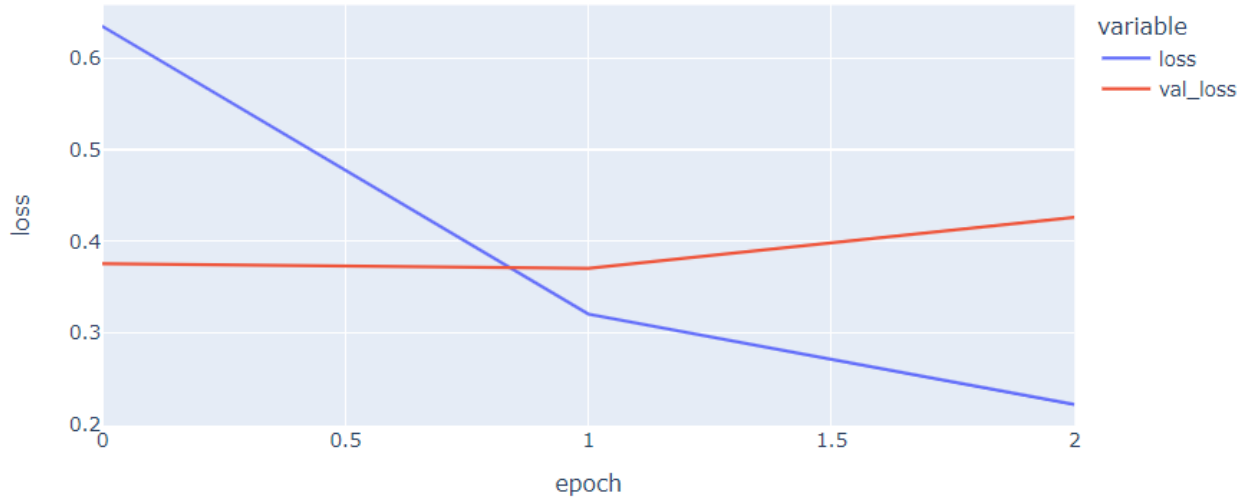
2 Dropout Layers

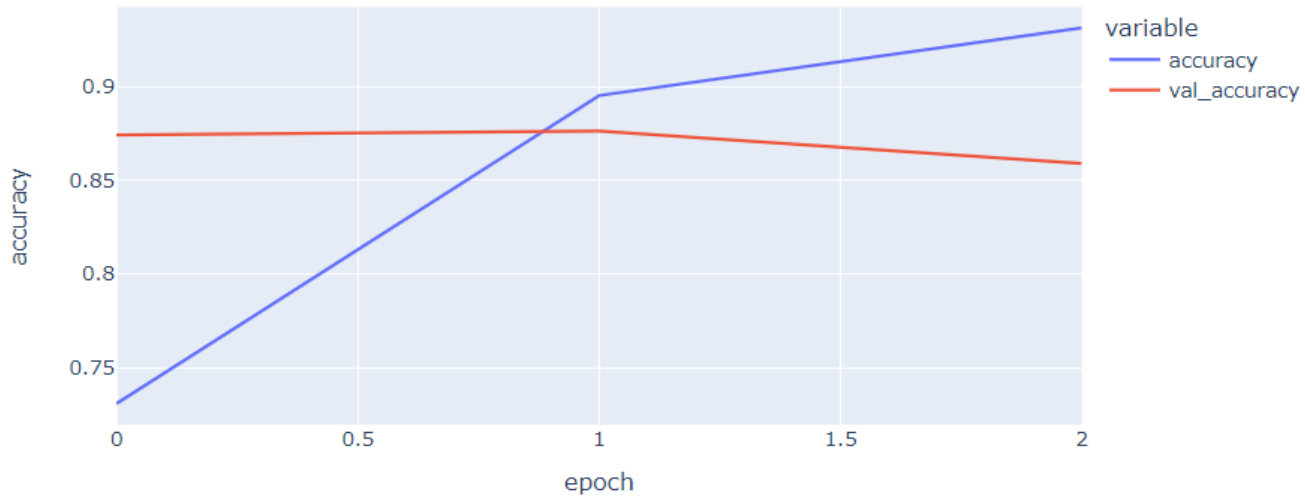
"Softmax" function in the output Layer
Optimizer = "ADAM"
Loss = "Categorical CrossEntropy"

Batch size = 32 and No. of Epochs = 3 and Verbose = 2.

The **final training accuracy** is = **85.81 %**
final testing accuracy is = **84.76 %**

And the final graphs for the training accuracy vs epoch and loss vs epoch is given below:





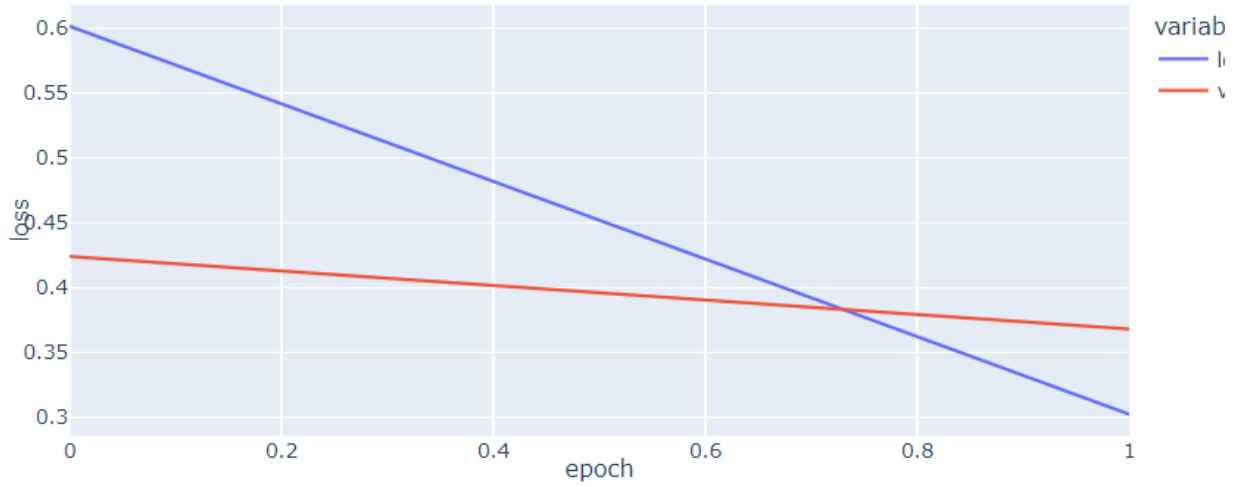
(b) Model a **<u>1 Layer LSTM with 64 hidden vector Dimension</u>** :

We used the different details for training the 1 Layer LSTM model , and these are as following :

Maximum features = 20000
Padding size maximum length = 200
Embedding Size = 128
Pooling Layer = “GlobalMaxPool”
2 Dense Layer
1 Dropout Layer
Optimizer = “ADAM”
Loss = “Categorical CrossEntropy”
Epochs = 2

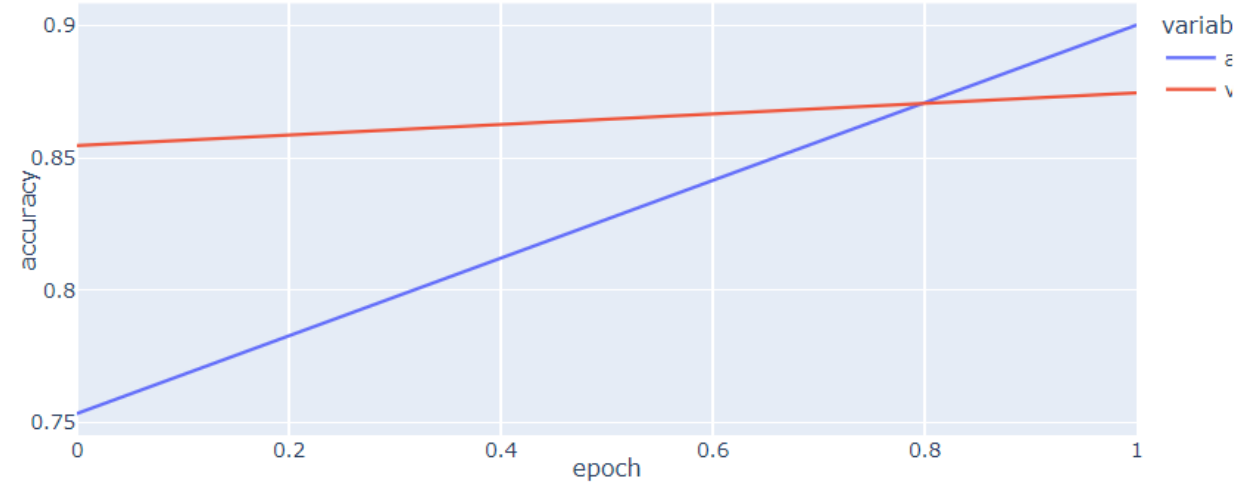
The **final training accuracy** is = **87.45 %**
final testing accuracy is = **87.454 %**

And the final graphs for the training accuracy vs epoch and loss vs epoch is given below:



”Red Line ” shows validation Loss

“Blue Line” shows training Loss



”Red Line ” shows validation Accuracy

“Blue Line” shows training Accuracy

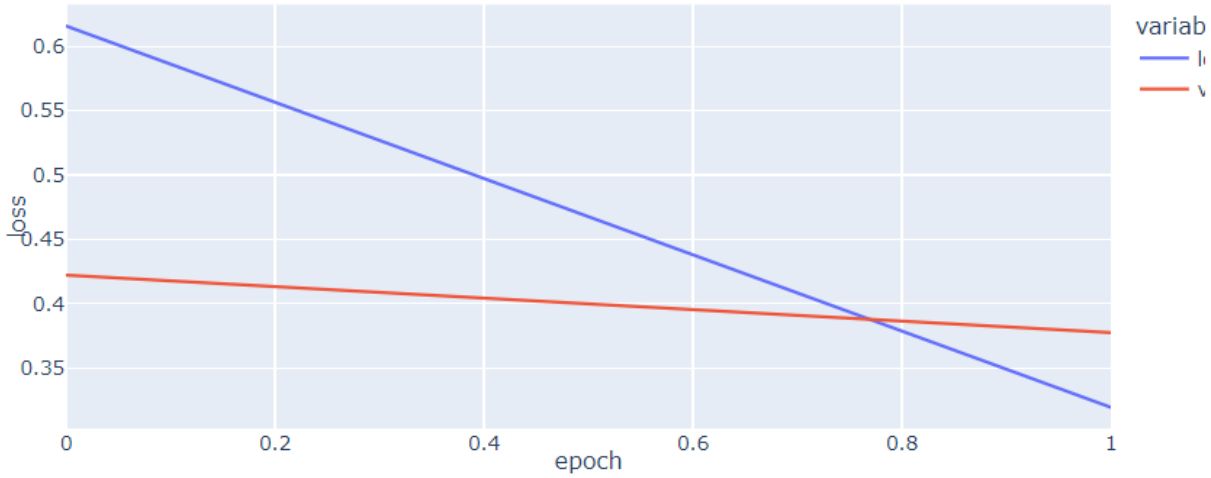
(c) Model a **<u>2 Layer LSTM with 64 hidden vector Dimension</u>** :

We used the different details for training the 1 Layer LSTM model , and these are as following :

Maximum features = 20000
Padding size maximum length = 200
Embedding Size = 128
Pooling Layer = “GlobalMaxPool”
2 Dense Layer
1 Dropout Layer
Optimizer = “ADAM”
Loss = “Categorical CrossEntropy”
Epochs = 2

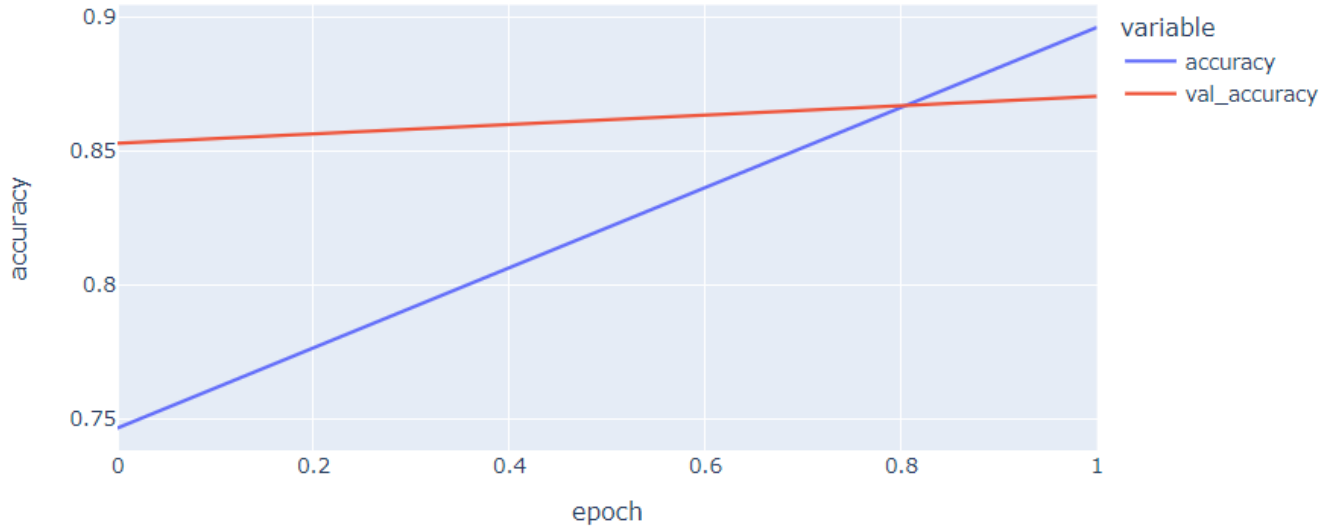
The **final training accuracy** is = **87.05 %**
final testing accuracy is = **87.053 %**

And the final graphs for the training accuracy vs epoch and loss vs epoch is given below:



”Red Line ” shows validation Loss

“Blue Line” shows training Loss



”Red Line ” shows validation Accuracy

“Blue Line” shows training Accuracy

(d) Model a **<u>Bi-LSTM with 64 hidden vector Dimension</u>** :

We used the different details for training the 1 Layer LSTM model , and these are as following :

Epochs = 2

Batch size = 32

Embedded Dimension = 16

Pooling Layer = “GlobalMaxPool”

2 Dense Layer

2 Dropout Layer

Optimizer = “ADAM”

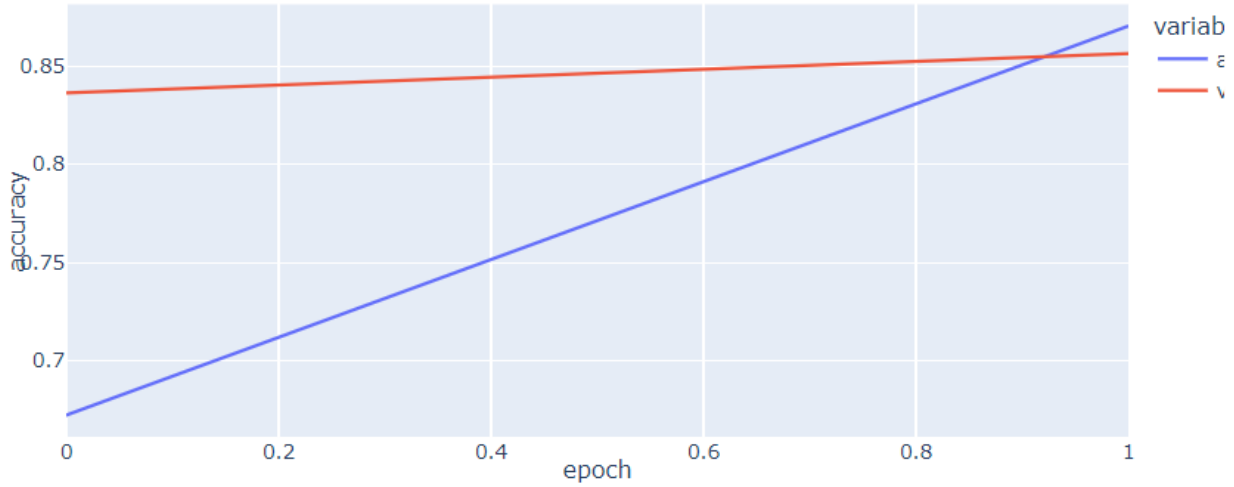
Loss = “Categorical CrossEntropy”

Activation Function : “Relu”

The **final training accuracy** is = **84.79 %**

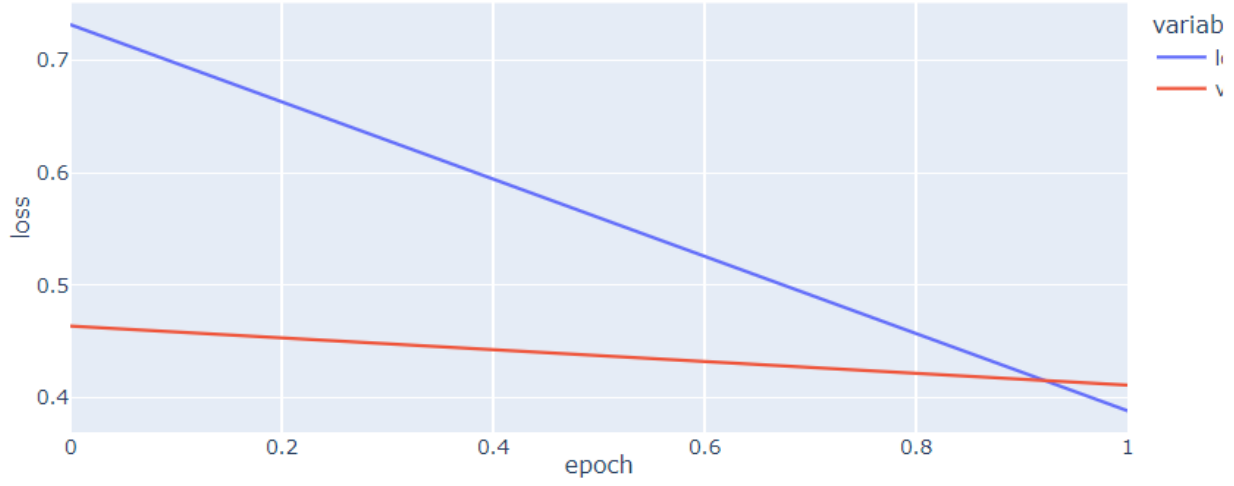
final testing accuracy is = **83.15 %**

And the final graphs for the training accuracy vs epoch and loss vs epoch is given below:



”Red Line ” shows validation Accuracy

“Blue Line” shows training Accuracy



”Red Line ” shows validation Loss

“Blue Line” shows training Loss

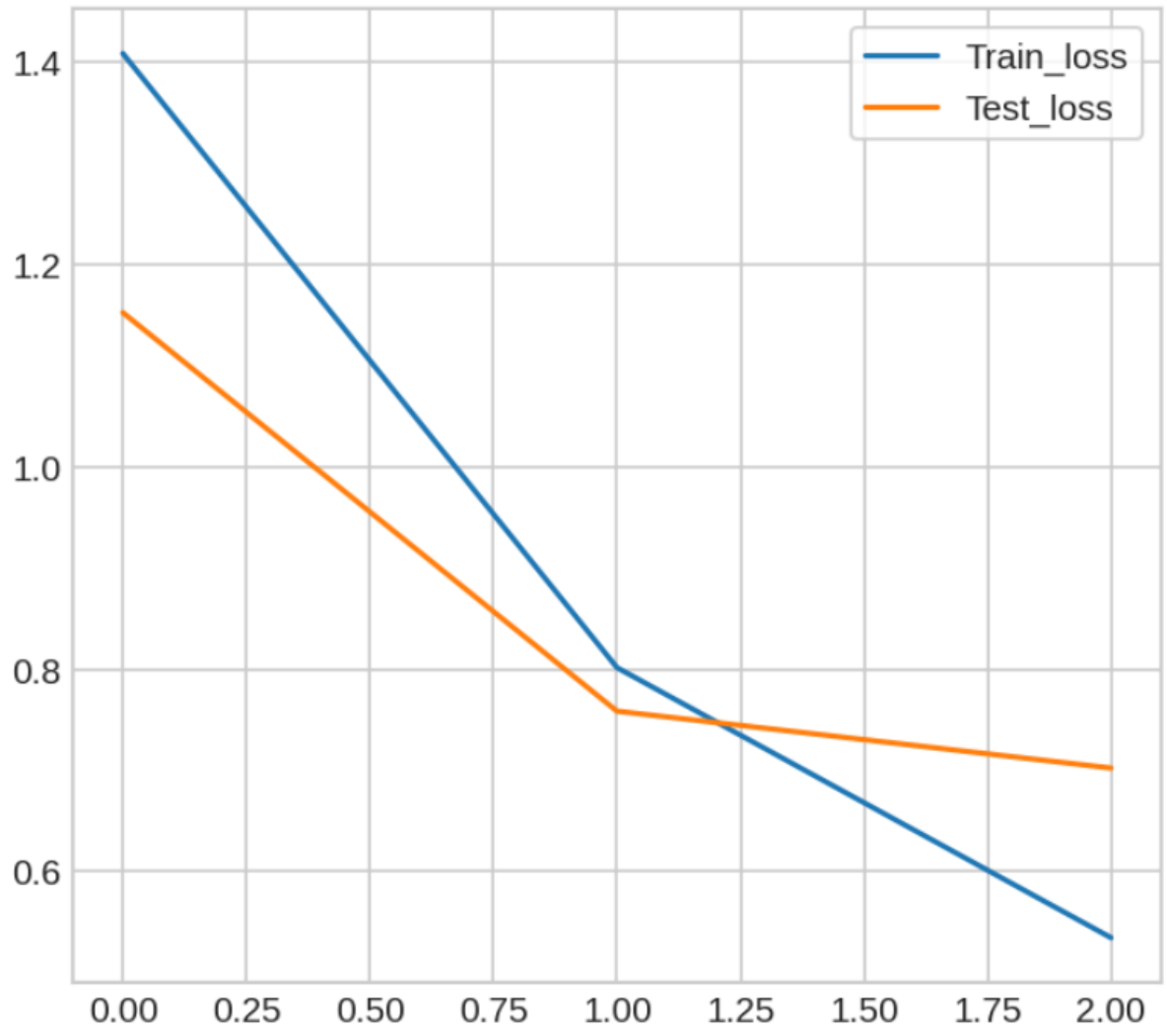
(f) <u>Use Glove embeddings as input embedding to model in 4.d.</u> :

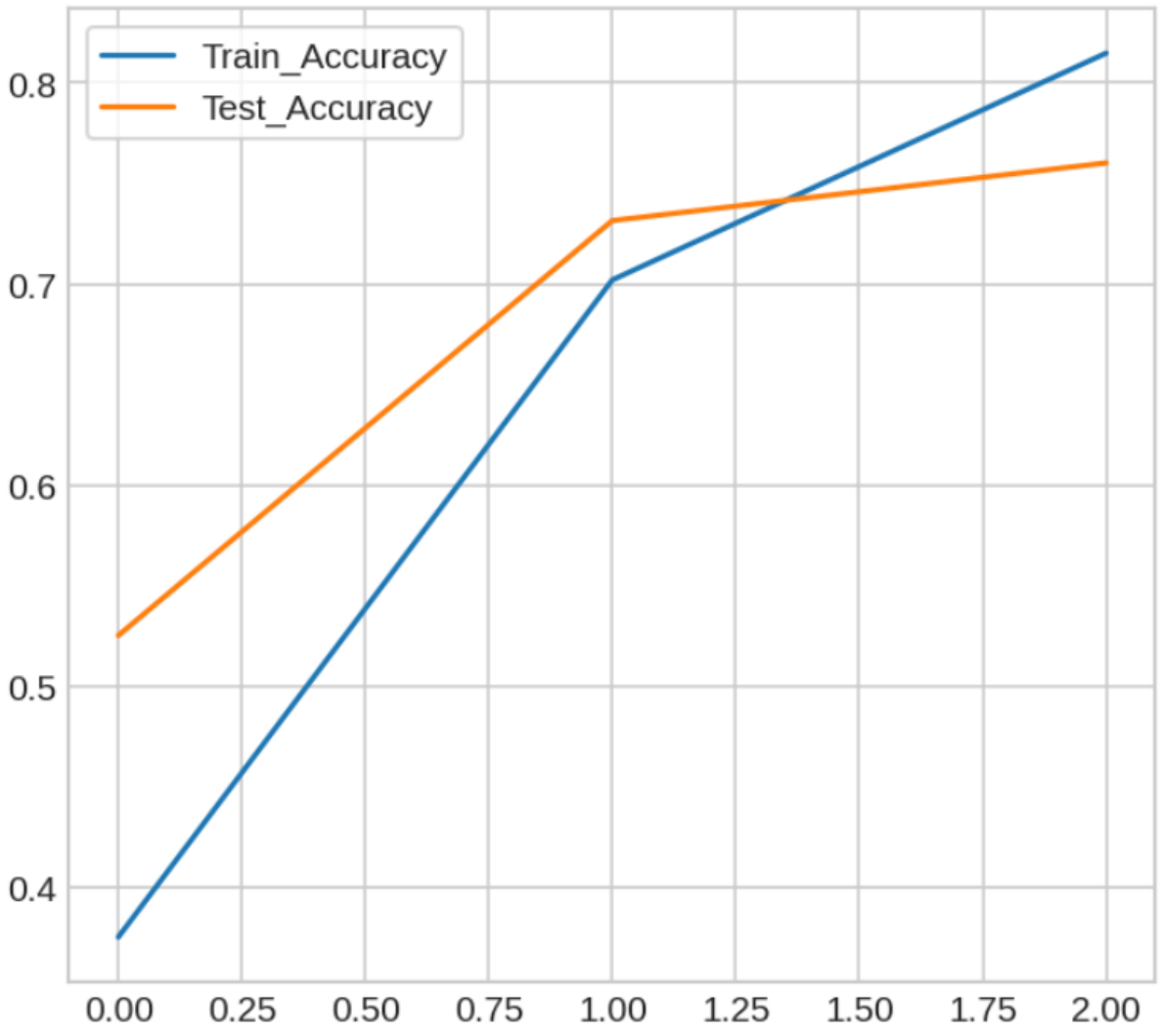
We used the different details for training the given model , and these are as following :

Maximum Sentence Length = 200
Batch size = 128
Embedding Dimension = 300
Number of Epochs = 3
Learning Rate = 0.0003
Optimizer = “ADAM”
Loss = “Categorical CrossEntropy”
Activation Function : “Relu”
2 Dropout Layer

The **final training accuracy** is = **81.47 %**
final testing accuracy is = **76.01 %**

And the final graphs for the training accuracy vs epoch and loss vs epoch is given below:


Train_loss
Test_loss
1.4
1.2
1.0
0.8
0.6
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00



Challenges Faced :

(i) Working on large data created a mess with the Compilation Part .

(ii) Due to Huge Computation we Run only Two or Three Epochs Only.

Contribution of each Group Member :

We contributed Equally in this Assignment , yet we are in including our contribution separately as instructed in the assignment

(i) Aditya Mishra (M20MA201) :

I trained all the models with the help of Atul and found the different accuracies again with the help of Atul .

I Plotted the different Plots of Accuracies and the losses of the training part.

I also resolved the error of the compilation with the help of Atul.

(ii) Atul Kumar Yadav (M20MA209) :

Atul Trained the Models with my Help also and found the different Errors in the compilation part and we did resolve them using each other's help.

We also supported each other in understanding the Algorithm of the Model we were working on.