UNIVERSITY OF ATHENS
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

Deep Learning for NLP

Student name: *Chatzispyrou Michail*
sdi: 1115202000212

Course: *Artificial Intelligence II (M138, M226, M262, M325)*
Semester: *Fall Semester 2023*

Contents

1. Abstract

This work studied the development of a sentiment classifier, using Feed Forward Neural Networks, for a twitter dataset regarding the Greek general elections. Each tweet consists of an ID, its actual text and lastly the class it belongs to: POSITIVE, NEUTRAL, NEGATIVE. The classifier handles all the classes. The experiments performed for this development uti- lized a variety of methods, including data processing, word embeddings and hyper-parameter tuning.

2. Data processing and analysis

2.1. Pre-processing

Pre-processing is an essential component of Artificial Intelligence (AI) and Machine Learning (ML), since it has the ability to convert raw input data into a more refined and useful form. Also, it can improve the quality and validity of the input data, which enhances both the efficiency and precision of the algorithms that come after.

An extensive data cleaning procedure was used to improve the dataset's quality and analytical appropriateness in order to create the current model. The act of removing links was crucial in getting rid of unnecessary site addresses and maintaining the dataset's primary emphasis. Similarly, by removing potential distractions from metadata, the systematic elimination of mentions and hashtags improved the dataset's readability. Text capitalization was standardized to lowercase in order to preserve uniformity throughout the dataset. Moreover, a further refinement phase required keeping only Latin or Greek-language words in order to guarantee linguistic consistency and remove non-alphabetic characters. Lemmatization was included as an additional option to the data cleaning process to further simplify the dataset in order to enable a more accurate and comprehensive analysis in the present study. Additionally, the removal of stopwords improved dataset conciseness by excluding common words with limited analytical value. Finally, accents were removed to ensure uniformity in text representation.

In the previous assignment, we investigated different pre-processing methods to improve our models' performance. We looked into three distinct strategies:

- No Pre-Processing (NP), the dataset was intact
- Basic Pre-Processing (BP), the dataset underwent the following modifications:
  - Removing links
  - Removing hashtags
  - Removing mentions
  - Removing stopwords
  - Removing accents
  - Lowercase
  - Removing non Latin or Greek-language words
- Basic Pre-Processing & Lemmatization (PL), all the Basic Pre-Processing with the addition Lemmatization

Following extensive testing, it became clear that, of the three approaches, Basic Pre-Processing (BP) produced the most encouraging outcomes. These results have led us to decide to simplify our strategy for this task. In order to preserve uniformity and enable a more precise comparison amongst models, we will just utilize the fundamental pre-processing method.

2.2. Analysis

We use a variety of tools to navigate the data's complexities as we begin our initial exploration phase and reveal its mysteries. Printing the dataframe (1) itself is the first step that is both straightforward and essential. Having a basic knowledge of the dataset's structure from this first look lets us examine its elements, column names, and the type of data it contains.

	New_ID	Text	Sentiment	Party
0	35027	#απολυμανση_κοριοι #απεντομωση_κοριος #απολυμανσεις #κοριος#Alphatv #Την Κυριακη #Κουλης #Τσιπρ...	NEUTRAL	SYRIZA
1	9531	Έξι νέες επιστολές για τη Μακεδονία «καίνε» τη ΝΔ - Ο Μητσοτάκης γνώριζε και δίχασε το έθνος htt...	NEGATIVE	ND
2	14146	Ισχυρό ΚΚΕ, δύναμη του λαού στη Βουλή και στους καθημερινούς αγώνες	POSITIVE	KKE
3	28716	@five2nds @anthi7vas Μνημονιακότατο το #ΜεΡΑ25 #Εκλογες_2019 #8iouliou #epomeni_mera #ΤΩΡΑ_ΚΚΕ ...	NEUTRAL	KKE
4	32886	@ai_katerina Αυτό που είναι συγκλονιστικό είναι η ψυχασθένεια του Τσίπρα!	NEUTRAL	SYRIZA
...	...	...	...	...
36625	35374	@KourtakisJohn @kmitsotakis Ο Κούλης ο Μητσοτάκης λέει ψέματα!!!Δεν άδειασε κανένα Μπάμπη Παπαδη...	NEUTRAL	ND
36626	7744	@enikos_gr @NChatzinikolaou @AdonisGeorgiadi Πρόσεξε μην σκίσει και κανένα καλσόν. Επίσης ... χά...	NEGATIVE	ND
36627	35216	Η θέση του ΚΚΕ για την ασφάλεια των πολιτών και τους διάφορους Ρουβίκωνες είναι η βαθιά κρίση το...	NEUTRAL	KKE
36628	2855	@thanosplevris Μαρη κακομοίρα θυγατέρα του ναζιστη αντισημίτη, έχεις ξεφτιλιστεί τόσο πολύ που ο...	NEGATIVE	ND
36629	25500	@gijjstalking @SpirosR76 Εντάξει με έπεισες! Και εγω ΚΚΕ! ✌️❤️	NEUTRAL	KKE

36630 rows × 4 columns

Figure 1: Train DataFrame

A snapshot of our dataset, consisting of a significant 36630 rows and carefully arranged into 4 columns, appears as we print the dataframe (1). Every tweet is given a unique identity by the first column. The tweets themselves are visible in the second column, which provides an insight into the textual world. Each tweet is a story contained inside the boundaries of the dataset. Furthermore, in the third column, we find the sentiment classes of neutral, positive, and negative. The last touch is found in the fourth column, where each tweet is matched with a particular political party based on its political resonance.

Next, we turn our attention to visualization, creating a bar plot (2) that provides a broad overview of the political fields included in the dataset. This visual aid offers a brief overview of how tweets are distributed among different political parties.

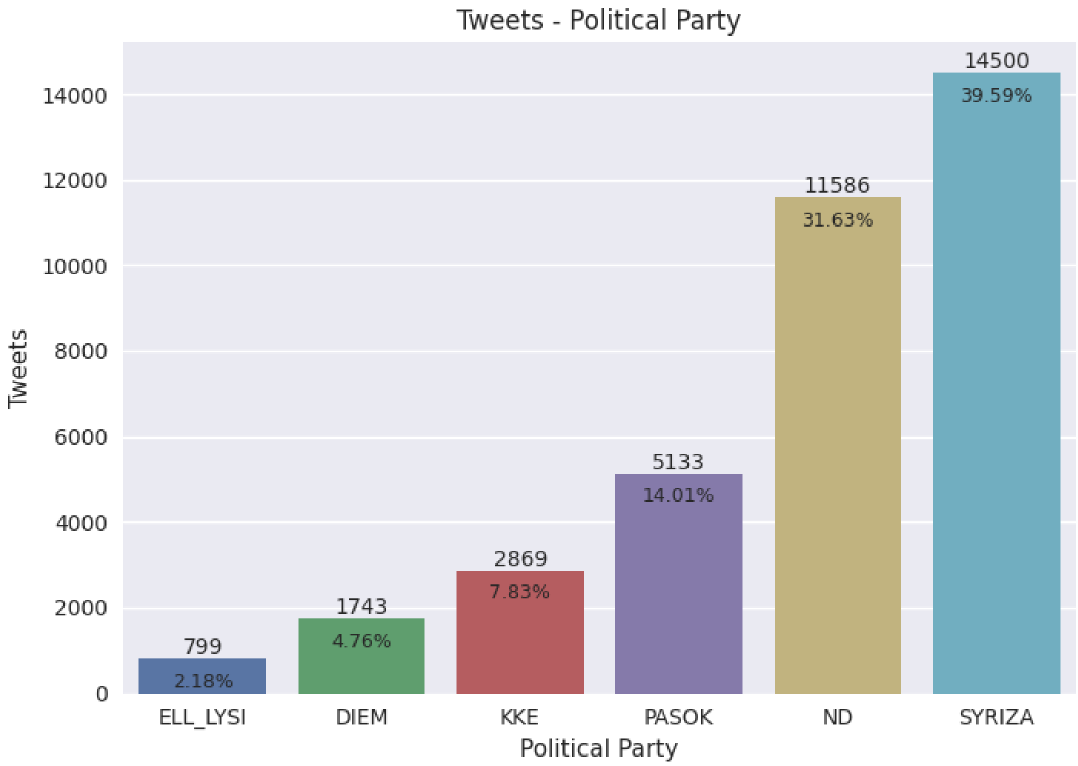

Figure 2: Tweets-Political Party Bar Plot

It can be observed that about 40% of the tweets examined deal with topics related to the political party called "SYRIZA". The New Democracy (ND) party comes in second, taking about 32% of the conversation on Twitter. The collection also contains references to "ELL_LYSI", which accounts for a little over 2% of the total and "PASOK," which accounts for a noteworthy 14%. Remarkably, the collection also includes references to "DIEM," but with a lesser frequency—less than 5%. At roughly 8%, the Communist Party of Greece (KKE) holds a significant portion.

To further our investigation, we create two separate bar plots, each of which highlights a different aspect of our dataset. The first bar plot (3) provides an in-depth comprehension of the distribution of neutral, positive, and negative attitudes inside each individual political party. At the same time, the second bar plot (4) turns our attention to a wider view, showing how tweets are distributed throughout the dataset in various sentiment categories.

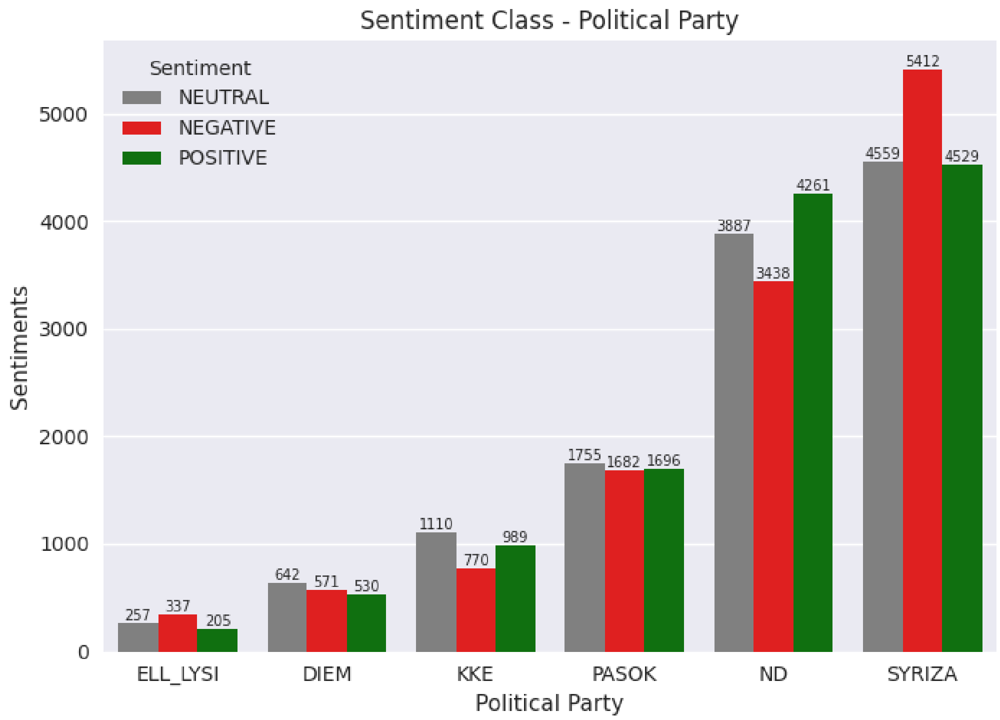

Figure 3: Sentiments-Political Party Bar Plot

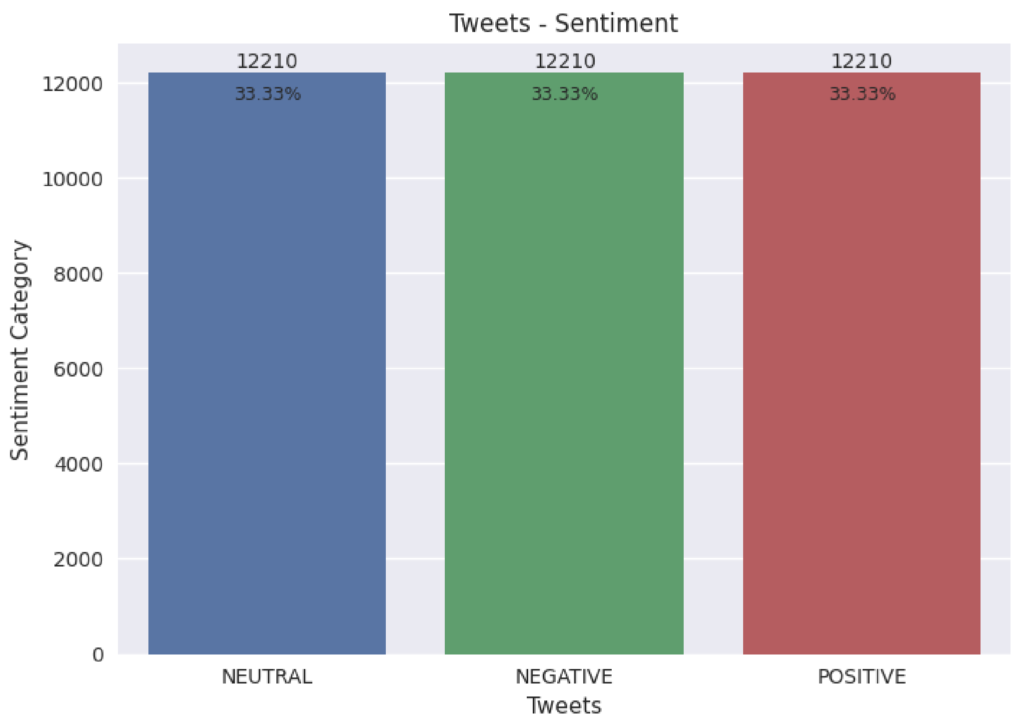

Figure 4: Tweets-Sentiment Bar Plot

Analysis reveals that while sentiment categories within each political party are not exactly balanced, they are admirably close to being balanced. This adds to an overall balanced dataset along with a fairly distributed distribution of tweets within every sentiment category. Fairness and ease of model training are two benefits of a balanced dataset. Unbiased exposure to every class ensures that no class has an undue influence on models during training. Finally, evaluation criteria that facilitate easy interpretation, such as accuracy, become more trustworthy measures of model performance.

2.3. Data partitioning for train, test and validation

The data sets chosen were the ones that the instructors provided. This decision was made in order to prevent any further complications regarding the development of the code. A ratio of 0.156 is attained by utilizing data from the validation set (5232) and training set (33630).

2.4. Vectorization

In our previous assignment, we looked at a variety of text representation methods to convert textual data into numerical formats appropriate for machine learning applications, including the TF-IDF vectorizer, CountVectorizer, and HashVectorizer. Every approach has advantages and disadvantages of its own, providing different insights on how best to convey the main ideas of the supporting material.

However for this project, we've gone in a new direction by using word embeddings in our study. Dense vector representations of words in a continuous vector space, where words with related meanings are positioned closer to one another, are called word embeddings. Word embeddings, in contrast to conventional vectorization algorithms, capture contextual information and semantic relationships, offering a deeper understanding of the underlying language.

We trained our model for this task using the well-known word embedding method, word2vec. Word2vec uses a given corpus's co-occurrence patterns to learn distributed representations of words. The end product is a 300-dimensional vector that captures the syntactic context and semantic meaning of each word.

We first tokenized the tweets in the pre-processing phase of our research, breaking them into individual words or tokens to make subsequent analysis easier. Building on this foundation, we have now employed a technique that makes use of the tokenization procedure to determine the mean vector for every tweet. Using the word embeddings that the word2vec model yields, we compute the mean vector for each tweet in the dataset. The mean vector is calculated by averaging the vectors representing each word in the tweet. Using this technique, we can create a representative vector for every tweet that captures all of the words' combined semantic information. These mean vectors are then saved in our dataframe.

	New_ID	Text	Sentiment	Party	Tokens	embeddings_mean_vector
0	35027	κυριακη κοριοι απολυμανση καταπολεμηση κοριων απεντομωση κοριους	NEUTRAL	SYRIZA	[κυριακη, κοριοι, απολυμανση, καταπολεμηση, κοριων, απεντομωση, κοριους]	[-0.32047054, -0.29309198, -0.20391123, 0.09438753, 0.20477511, -0.23843409, -0.124472566, 0.071...
1	9531	νεες επιστολες μακεδονια καινε νδ μητσοτακης γνωριζε διχασε εθνος	NEGATIVE	ND	[νεες, επιστολες, μακεδονια, καινε, νδ, μητσοτακης, γνωριζε, διχασε, εθνος]	[-0.23106793, -0.7853444, 0.22757609, 0.546944, -0.22676288, -0.4536085, -1.3300469, -0.2669274,...
2	14146	ισχυρο κκε δυναμη λαου βουλη καθημερινους αγωνες	POSITIVE	KKE	[ισχυρο, κκε, δυναμη, λαου, βουλη, καθημερινους, αγωνες]	[-0.44802904, 0.9100896, 0.2015739, 0.012176646, -0.62373096, -0.091203734, -0.1007744, 0.360764...
3	28716	μνημονιακοτατο	NEUTRAL	KKE	[μνημονιακοτατο]	[-0.0010097063, -0.0009839003, 0.001484412, 0.00292768, 0.0013669952, -0.0021932323, -0.00275025...
4	32886	συγκλονιστικο ψυχασθενεια τσιπρα	NEUTRAL	SYRIZA	[συγκλονιστικο, ψυχασθενεια, τσιπρα]	[0.005598699, 0.17969884, -0.08877006, 0.03008213, 0.35266855, -0.6981933, -0.28598562, -0.04011...
...	...	...	...	...	...	...
36625	35374	κουλης μητσοτακης λεει ψεματα αδειασε μπαμπη παπαδημητριου μπαμπης προετοιμαζει ερχεται κουλη μη...	NEUTRAL	ND	[κουλης, μητσοτακης, λεει, ψεματα, αδειασε, μπαμπη, παπαδημητριου, μπαμπης, προετοιμαζει, ερχετα...	[-0.11121101, -0.25207365, 0.28019226, 0.4195406, -0.16259795, -0.7871607, -0.9632984, 0.5553633...
36626	7744	προσεξε σκισει καλσον χαλια νεα φωτογραφια ομορφη πραγματικοτητα	NEGATIVE	ND	[προσεξε, σκισει, καλσον, χαλια, νεα, φωτογραφια, ομορφη, πραγματικοτητα]	[-0.0568525, -0.08007222, -0.13146874, 0.23573276, 0.07861893, -0.37239, -0.39985576, 0.00115527...
36627	35216	θεση κκε ασφαλεια πολιτων διαφορους ρουβικωνες βαθια κριση καπιταλιστικου συστηματος μειωση φορο...	NEUTRAL	KKE	[θεση, κκε, ασφαλεια, πολιτων, διαφορους, ρουβικωνες, βαθια, κριση, καπιταλιστικου, συστηματος, ...	[-0.16773018, 0.040738173, -0.0970156, -0.10762447, -0.25490364, -0.0043183854, -0.22932982, -0....
36628	2855	μαρη κακομοιρα θυγατερα ναζιστη αντισημιτη ξεφτιλιστει τρολ νδ σχολιαζουν ντροπη μπορεις λιποθυμ...	NEGATIVE	ND	[μαρη, κακομοιρα, θυγατερα, ναζιστη, αντισημιτη, ξεφτιλιστει, τρολ, νδ, σχολιαζουν, ντροπη, μπορ...	[-0.10868633, -0.14007707, 0.05934826, 0.04757132, -0.15126355, -0.12087498, -0.34052882, -0.185...
36629	25500	ενταξει επεισες κκε	NEUTRAL	KKE	[ενταξει, επεισες, κκε]	[-0.022761554, 0.1834997, 0.59880704, -0.08725914, -0.45202366, 0.17532952, 0.035075385, 0.11981...

36630 rows × 6 columns

Figure 5: Train DataFrame with Pre-Processing & Word2Vec

3. Algorithms and Experiments

The classification task in the conducted experiments was carried out with an emphasis on grouping data into three separate classes. A variety of feed-forward neural networks with distinct architectures and configurations were utilized. The main goal was to assess how neural network architecture affected the classification accuracy for each of the specified classes.

3.1. Hyper-parameter tuning

The architecture and training dynamics of feedforward neural networks are significantly shaped by a variety of hyperparameters. The network can detect complex patterns in the data thanks to the non-linearity introduced by the activation function selection. Each hidden layer's layer size controls how many neurons are present in a particular layer, which affects the network's ability to learn complex representations. In order to avoid overfitting and improve generalization, dropout rate, a type of regularization, randomly deactivates a portion of neurons during training. The training algorithm that modifies the weights is determined by the optimizer. The depth of the network, which affects its capacity to represent hierarchical information, is determined by the number of hidden layers. It is crucial to give these hyperparameters careful thought and tuning in order to ensure that the performance of the neural network is optimal.

3.2. Evaluation

A thorough set of six measures, each providing a unique perspective on the model's performance, will be included in the evaluation process. These measures are f1-score, accuracy, recall, precision, average loss and time taken. the F1-score is given particular consideration since it may effectively balance recall and precision. The F1-score is calculated as follows:

$$F1\ Score = 2 \times \frac{precision \times recall}{precision + recall} \tag{1}$$

In addition, a Confusion Matrix and a Receiver Operating Characteristic (ROC) curve were used for more in-depth study of the top model found using these measures.

3.3. Experiments

Our goal in this project is to investigate and choose the best neural network architecture for our dataset while balancing the restricted amount of data we have with the complexity of the models. We will concentrate on three distinct architectures, with one to three hidden layers, since going over this limit could lead to overfitting because of the lack of available data. We will change the layer sizes inside each architecture to see how it affects the performance of the model. We will not use activation functions or dropout ratios in order to keep the first experimentation as straightforward as possible. A standard set of hyperparameters will be used in the experiments, such as a learning rate of 0.01, the optimizer SGD and the loss function CrossEntropyLoss. Every model will undergo 50 training epochs. Learning curves (loss and F1 scores) during training, along with thorough classification reports for the training and validation sets, will be the basis for evaluation. After then, the most promising design will be further improved by experimenting with various optimizers, dropout mechanisms, and activation functions. This exhaustive method seeks to methodically identify an architecture that applies well to our particular dataset.

3.3.1. Architecture 1.

The first architecture will be consisted of the following:

- InputDimension: 300
- OneHiddenLayer: [16, 32, 64, 128, 256]
- OutputDimension: 3

——————- Neural Network with 1 Hidden Layer of size 16 ——————

Time Taken:	134.649
Average Loss:	1.0822728748350698
F1-Score:	0.3675421476364136
Accuracy:	0.3864678740501404
Recall:	0.3864678740501404
Precision:	0.3905011117458343

Table 1: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.37	0.58	0.45	12210
NEUTRAL	0.41	0.18	0.25	12210
POSITIVE	0.40	0.39	0.40	12210
accuracy			0.39	36630
macro avg	0.39	0.39	0.37	36630
weighted avg	0.39	0.39	0.37	36630

Table 2: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.38	0.58	0.46	1744
NEUTRAL	0.40	0.18	0.25	1744
POSITIVE	0.40	0.40	0.40	1744
accuracy			0.39	5232
macro avg	0.39	0.39	0.37	5232
weighted avg	0.39	0.39	0.37	5232

Table 3: Val Classification Report

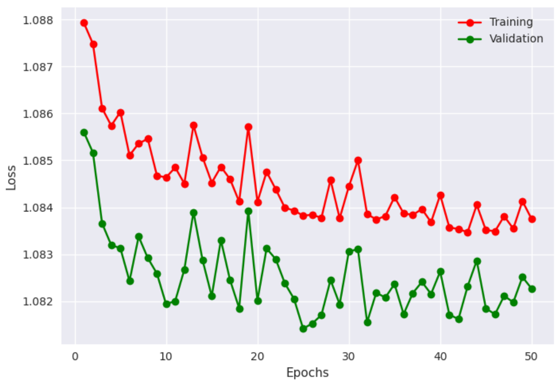

Figure 6: Learnig Curve (Loss)

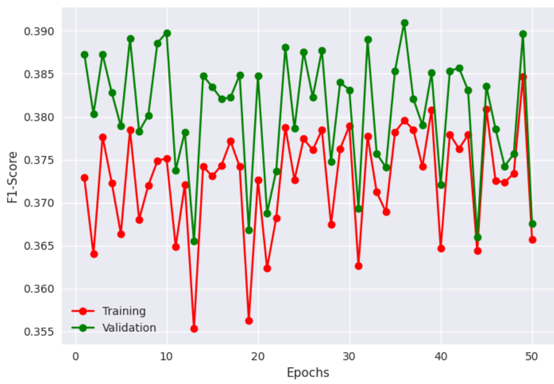

Figure 7: Learnig Curve (F1-Score)

Neural Network with 1 Hidden Layer of size 32

Time Taken:	143.558
Average Loss:	1.0822088915273684
F1-Score:	0.3726451992988586
Accuracy:	0.3878058195114136
Recall:	0.3878058195114136
Precision:	0.3904195427894592

Table 4: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.37	0.59	0.45	12210
NEUTRAL	0.40	0.21	0.27	12210
POSITIVE	0.41	0.37	0.39	12210
accuracy			0.39	36630
macro avg	0.39	0.39	0.37	36630
weighted avg	0.39	0.39	0.37	36630

Table 5: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.38	0.58	0.46	1744
NEUTRAL	0.39	0.21	0.27	1744
POSITIVE	0.40	0.37	0.39	1744
accuracy			0.39	5232
macro avg	0.39	0.39	0.37	5232
weighted avg	0.39	0.39	0.37	5232

Table 6: Val Classification Report

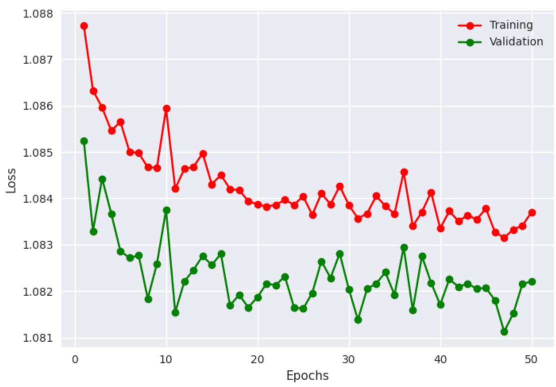

Figure 8: Learnig Curve (Loss)

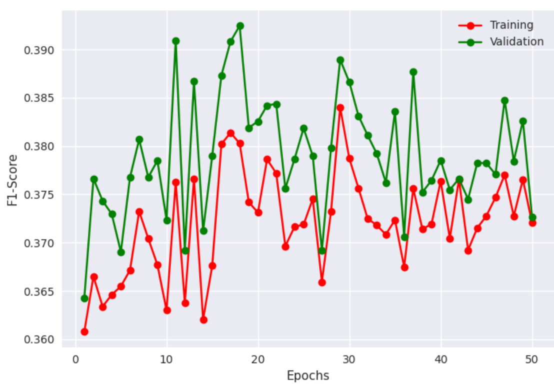

Figure 9: Learnig Curve (F1-Score)

Neural Network with 1 Hidden Layer of size 64

Time Taken:	145.285
Average Loss:	1.0822704804417556
F1-Score:	0.3824034035205841
Accuracy:	0.3910550475120544
Recall:	0.3910550475120544
Precision:	0.3909140825271606

Table 7: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.38	0.42	0.40	12210
NEUTRAL	0.39	0.24	0.30	12210
POSITIVE	0.38	0.48	0.43	12210
accuracy			0.38	36630
macro avg	0.38	0.38	0.37	36630
weighted avg	0.38	0.38	0.37	36630

Table 8: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.40	0.43	0.41	1744
NEUTRAL	0.39	0.24	0.30	1744
POSITIVE	0.39	0.51	0.44	1744
accuracy			0.39	5232
macro avg	0.39	0.39	0.38	5232
weighted avg	0.39	0.39	0.38	5232

Table 9: Val Classification Report

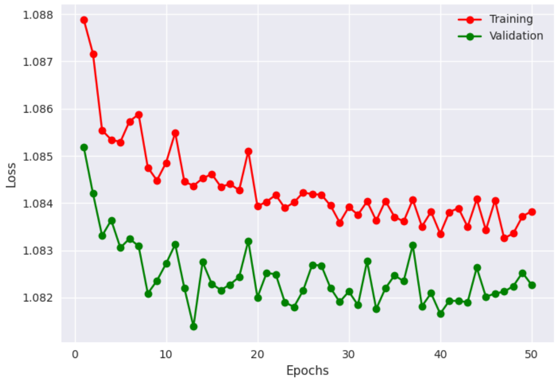

Figure 10: Learnig Curve (Loss)

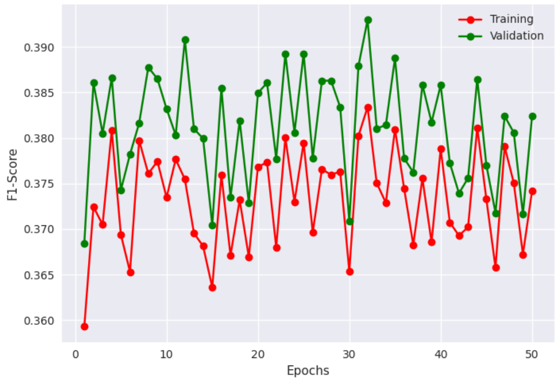

Figure 11: Learnig Curve (F1-Score)

Neural Network with 1 Hidden Layer of size 128

Time Taken:	148.533
Average Loss:	1.0817800960774087
F1-Score:	0.3770482242107391
Accuracy:	0.3912461698055267
Recall:	0.3912461698055267
Precision:	0.3925237059593200

Table 10: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.37	0.54	0.44	12210
NEUTRAL	0.39	0.20	0.26	12210
POSITIVE	0.40	0.42	0.41	12210
accuracy			0.38	36630
macro avg	0.39	0.38	0.37	36630
weighted avg	0.39	0.38	0.37	36630

Table 11: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.38	0.53	0.45	1744
NEUTRAL	0.39	0.20	0.27	1744
POSITIVE	0.40	0.44	0.42	1744
accuracy			0.39	5232
macro avg	0.39	0.39	0.38	5232
weighted avg	0.39	0.39	0.38	5232

Table 12: Val Classification Report

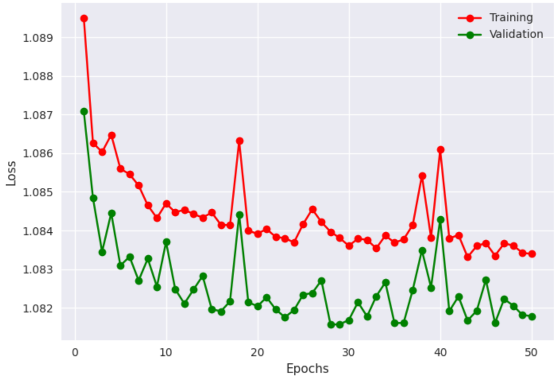

Figure 12: Learnig Curve (Loss)

Figure 13: Learnig Curve (F1-Score)

Neural Network with 1 Hidden Layer of size 256

Time Taken:	163.958
Average Loss:	1.081850341153801
F1-Score:	0.379727542400360
Accuracy:	0.387614667415618
Recall:	0.387614667415618
Precision:	0.391861379146575

Table 13: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.37	0.54	0.44	12210
NEUTRAL	0.37	0.31	0.34	12210
POSITIVE	0.42	0.30	0.35	12210
accuracy			0.38	36630
macro avg	0.39	0.38	0.37	36630
weighted avg	0.39	0.38	0.37	36630

Table 14: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.38	0.55	0.45	1744
NEUTRAL	0.38	0.31	0.34	1744
POSITIVE	0.42	0.30	0.35	1744
accuracy			0.39	5232
macro avg	0.39	0.39	0.38	5232
weighted avg	0.39	0.39	0.38	5232

Table 15: Val Classification Report

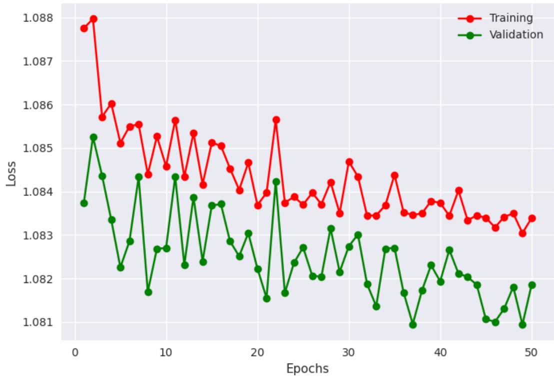

Figure 14: Learnig Curve (Loss)

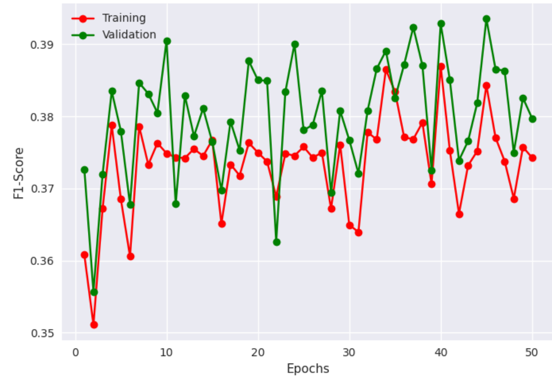

Figure 15: Learnig Curve (F1-Score)

Analysis of the first architecture

When it comes to neural networks with a single hidden layer, increasing the number of neurons can lead to longer training times. However, this does not necessarily translate to improved accuracy. The learning curves, which include both loss (14) and F1-score (15), can help us understand the model's performance and generalization ability. It is important to note that spikes in the learning curves do not always indicate a lack of generalization ability. In our cases, the values on the axis are too close together and in combination with the option not to use any activation function, leading to large spikes that do not necessarily reflect the model's performance. When comparing different neural networks within the same architecture, it is essential to consider the classification report, which provides insights into the model's performance in terms of accuracy, precision, recall, and F1-score. In this case, the neural network with 256 neurons appears to have a more evenly balanced F1-score between all three classes (14, 15), making it a suitable choice for the given problem.

3.3.2. Architecture 2.

The second architecture will be consisted of the following:

- InputDimension: 300
- FirstHiddenLayer: [16, 32, 64, 128, 256]
- SecondHiddenLayer: [16, 32, 64, 128, 256]
- OutputDimension: 3

In the second neural network architecture, there are 25 possible experiments due to the fact that each of the hidden layers has 5 possible sizes. To ensure that the report is as comprehensive as possible, I will present some of the experiments, but I will provide an explanation for all of them.

Neural Network with 2 Hidden Layer of sizes 16 16

Time Taken:	144.163
Average Loss:	1.0819223805669616
F1-Score:	0.3799508810043335
Accuracy:	0.3881880939006805
Recall:	0.3881880939006805
Precision:	0.3869755268096924

Table 16: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.37	0.52	0.43	12210
NEUTRAL	0.39	0.25	0.31	12210
POSITIVE	0.40	0.38	0.39	12210
accuracy			0.38	36630
macro avg	0.39	0.38	0.38	36630
weighted avg	0.39	0.38	0.38	36630

Table 17: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.39	0.52	0.44	1744
NEUTRAL	0.37	0.25	0.30	1744
POSITIVE	0.40	0.39	0.40	1744
accuracy			0.39	5232
macro avg	0.39	0.39	0.38	5232
weighted avg	0.39	0.39	0.38	5232

Table 18: Val Classification Report

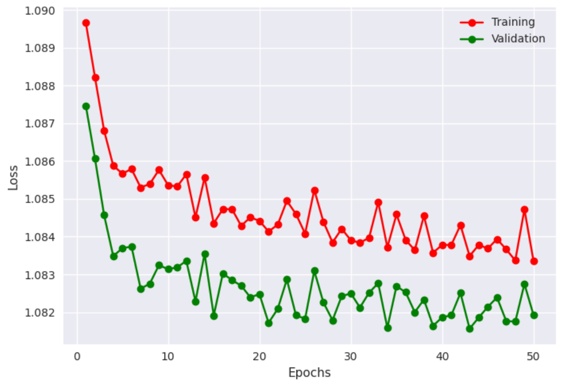

Figure 16: Learnig Curve (Loss)

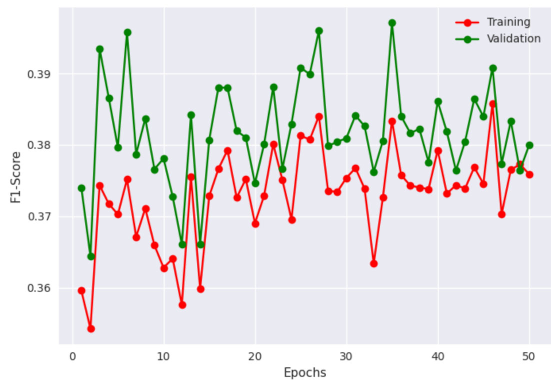

Figure 17: Learnig Curve (F1-Score)

Neural Network with 2 Hidden Layer of sizes 16 32

Time Taken:	142.979
Average Loss:	1.0819346979488291
F1-Score:	0.3762797713279724
Accuracy:	0.3902904987335205
Recall:	0.3902904987335205
Precision:	0.3905092477798462

Table 19: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.37	0.51	0.43	12210
NEUTRAL	0.39	0.20	0.27	12210
POSITIVE	0.39	0.43	0.41	12210
accuracy			0.38	36630
macro avg	0.38	0.38	0.37	36630
weighted avg	0.38	0.38	0.37	36630

Table 20: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.39	0.52	0.44	1744
NEUTRAL	0.39	0.20	0.26	1744
POSITIVE	0.39	0.46	0.42	1744
accuracy			0.39	5232
macro avg	0.39	0.39	0.38	5232
weighted avg	0.39	0.39	0.38	5232

Table 21: Val Classification Report

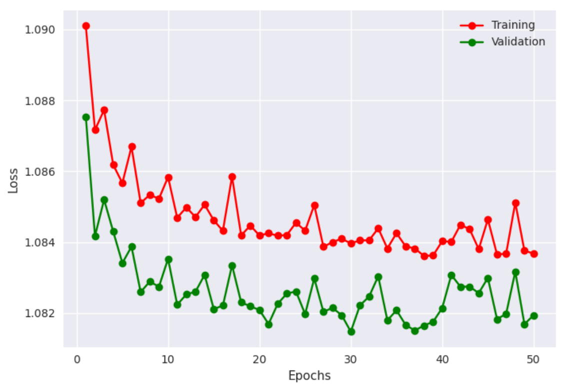

Figure 18: Learnig Curve (Loss)

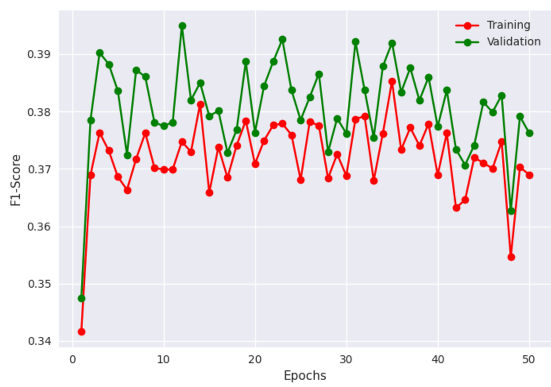

Figure 19: Learnig Curve (F1-Score)

Neural Network with 2 Hidden Layer of sizes 32 32

Time Taken:	153.715
Average Loss:	1.0819116283994201
F1-Score:	0.3852098882198334
Accuracy:	0.3904816508293152
Recall:	0.3904816508293152
Precision:	0.3896832764148712

Table 22: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.37	0.45	0.41	12210
NEUTRAL	0.38	0.27	0.31	12210
POSITIVE	0.39	0.43	0.41	12210
accuracy			0.38	36630
macro avg	0.38	0.38	0.38	36630
weighted avg	0.38	0.38	0.38	36630

Table 23: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.39	0.45	0.42	1744
NEUTRAL	0.38	0.27	0.32	1744
POSITIVE	0.40	0.45	0.42	1744
accuracy			0.39	5232
macro avg	0.39	0.39	0.39	5232
weighted avg	0.39	0.39	0.39	5232

Table 24: Val Classification Report

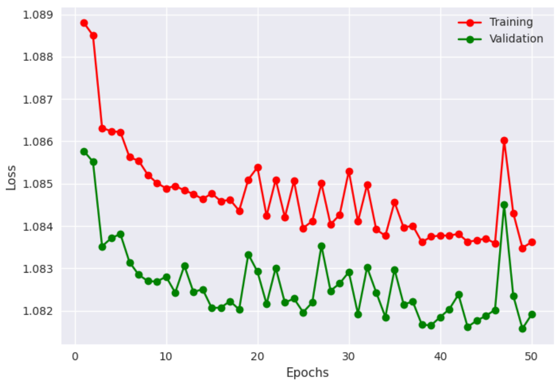

Figure 20: Learnig Curve (Loss)

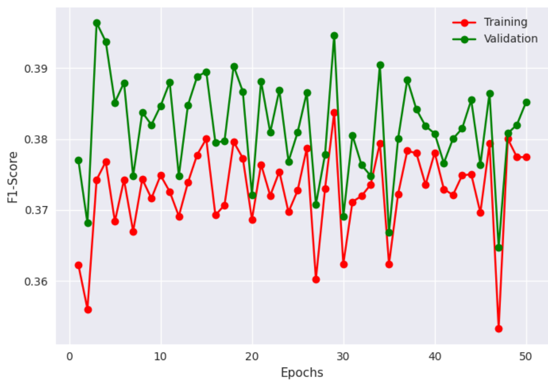

Figure 21: Learnig Curve (F1-Score)

Neural Network with 2 Hidden Layer of sizes 32 128

Time Taken:	155.415
Average Loss:	1.081572185962572
F1-Score:	0.379146695137023
Accuracy:	0.390672773122787
Recall:	0.390672773122787
Precision:	0.392389059066772

Table 25: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.37	0.57	0.45	12210
NEUTRAL	0.39	0.24	0.30	12210
POSITIVE	0.41	0.35	0.37	12210
accuracy			0.38	36630
macro avg	0.39	0.38	0.37	36630
weighted avg	0.39	0.38	0.37	36630

Table 26: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.38	0.57	0.46	1744
NEUTRAL	0.39	0.24	0.30	1744
POSITIVE	0.41	0.36	0.38	1744
accuracy			0.39	5232
macro avg	0.39	0.39	0.38	5232
weighted avg	0.39	0.39	0.38	5232

Table 27: Val Classification Report

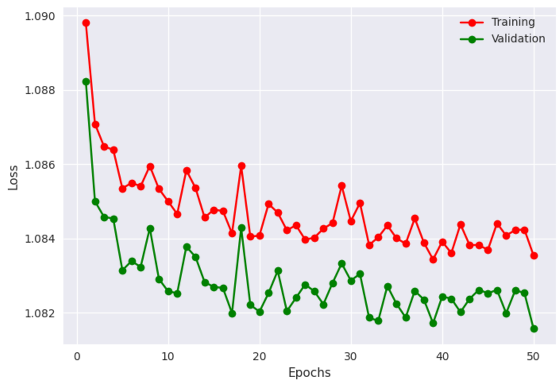

Figure 22: Learnig Curve (Loss)

Figure 23: Learnig Curve (F1-Score)

Neural Network with 2 Hidden Layer of sizes 64 128

Time Taken:	169.952
Average Loss:	1.0816798193739094
F1-Score:	0.3858848512172699
Accuracy:	0.3895260095596313
Recall:	0.3895260095596313
Precision:	0.3910441100597381

Table 28: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.37	0.50	0.43	12210
NEUTRAL	0.37	0.32	0.35	12210
POSITIVE	0.41	0.33	0.36	12210
accuracy			0.38	36630
macro avg	0.39	0.38	0.38	36630
weighted avg	0.39	0.38	0.38	36630

Table 29: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.38	0.50	0.43	1744
NEUTRAL	0.38	0.33	0.35	1744
POSITIVE	0.41	0.34	0.37	1744
accuracy			0.39	5232
macro avg	0.39	0.39	0.39	5232
weighted avg	0.39	0.39	0.39	5232

Table 30: Val Classification Report

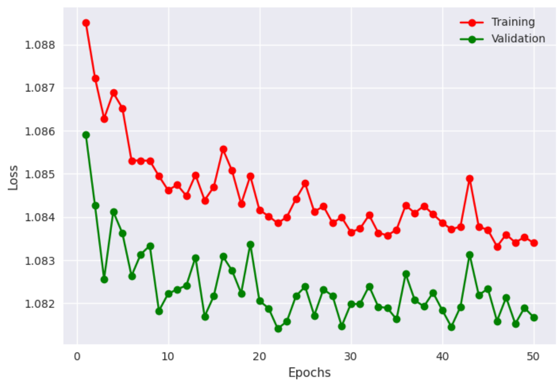

Figure 24: Learnig Curve (Loss)

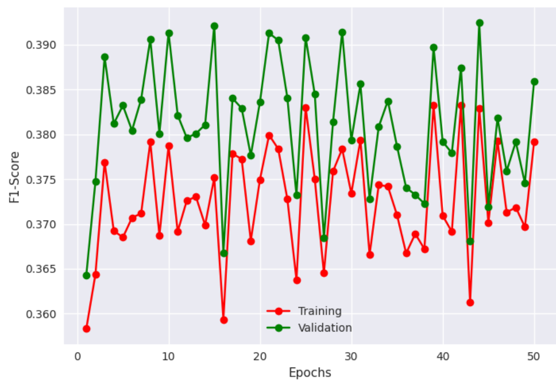

Figure 25: Learnig Curve (F1-Score)

Neural Network with 2 Hidden Layer of sizes 128 32

Time Taken:	162.588
Average Loss:	1.0816545305995766
F1-Score:	0.3820842504501343
Accuracy:	0.3920106887817383
Recall:	0.3920106887817383
Precision:	0.3922990560531616

Table 31: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.37	0.52	0.43	12210
NEUTRAL	0.39	0.24	0.29	12210
POSITIVE	0.40	0.39	0.39	12210
accuracy			0.38	36630
macro avg	0.39	0.38	0.37	36630
weighted avg	0.39	0.38	0.37	36630

Table 32: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.38	0.53	0.44	1744
NEUTRAL	0.39	0.23	0.29	1744
POSITIVE	0.41	0.41	0.41	1744
accuracy			0.39	5232
macro avg	0.39	0.39	0.38	5232
weighted avg	0.39	0.39	0.38	5232

Table 33: Val Classification Report

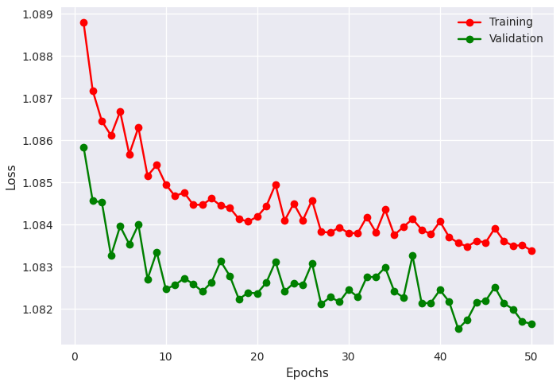

Figure 26: Learnig Curve (Loss)

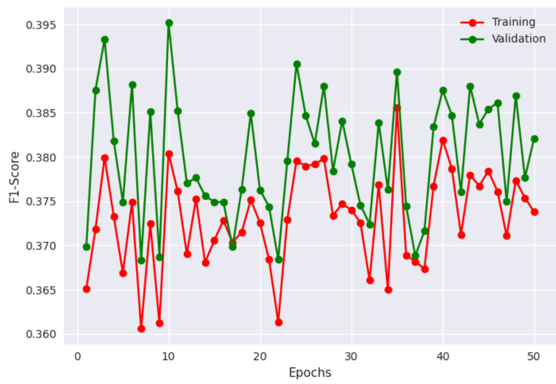

Figure 27: Learnig Curve (F1-Score)

Neural Network with 2 Hidden Layer of sizes 256 64

Time Taken:	191.655
Average Loss:	1.081967358560008
F1-Score:	0.391902267932891
Accuracy:	0.393157482147216
Recall:	0.393157482147216
Precision:	0.392814993858337

Table 34: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.38	0.39	0.39	12210
NEUTRAL	0.38	0.34	0.36	12210
POSITIVE	0.39	0.42	0.41	12210
accuracy			0.39	36630
macro avg	0.39	0.39	0.39	36630
weighted avg	0.39	0.39	0.39	36630

Table 35: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.40	0.40	0.40	1744
NEUTRAL	0.39	0.34	0.36	1744
POSITIVE	0.39	0.45	0.42	1744
accuracy			0.39	5232
macro avg	0.39	0.39	0.39	5232
weighted avg	0.39	0.39	0.39	5232

Table 36: Val Classification Report

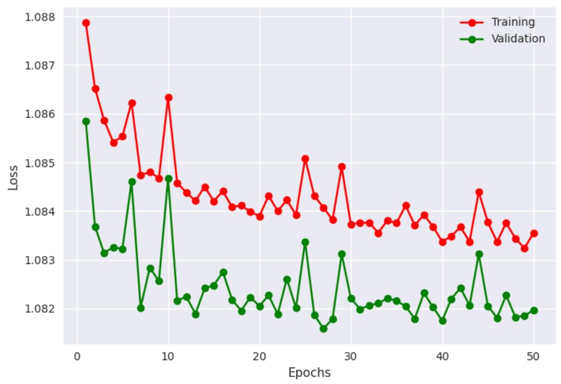

Figure 28: Learnig Curve (Loss)

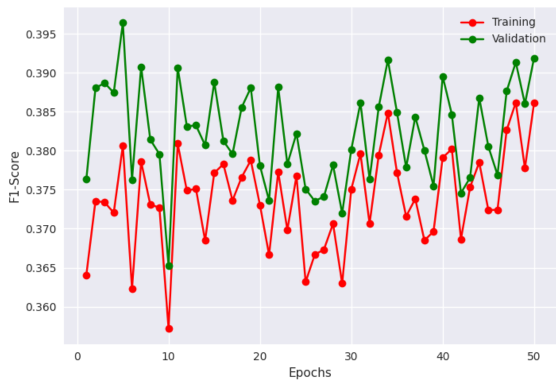

Figure 29: Learnig Curve (F1-Score)

Analysis of the second architecture

When examining the second neural network architecture, we can observe that the time required for training varies depending on the number of neurons in the first and second hidden layers. Specifically, when the first hidden layer has 16 neurons, the time required for training increases as the number of neurons in the second hidden layer increases, up until 32 neurons. However, the time required for training decreases for the 64 and 128 neuron configurations, before increasing again for the 256 neuron configuration. The distribution of F1-scores is much better in this architecture compared to the first architecture, indicating improved classification performance. For the first hidden layer with 32 neurons, we can observe a steady increment in time required for training as the size of the second hidden layer increases. However, the classification reports for these configurations are not as promising as those for the 16 neuron configuration in the first hidden layer. When the first hidden layer has 64 neurons, the time and accuracy are stable and low for configurations where the second hidden layer has the same size or fewer neurons. However, when the second hidden layer has more neurons, both the time and metrics increase. For the first hidden layer with 128 neurons, the time required for training increases as the number of neurons in the second hidden layer increases, but the metrics decrease. Finally, for the first hidden layer with 256 neurons, we observe a similar pattern as before, with the time required for training increasing as the size of the second layer increases. However, up until the size of 64, the accuracy is increasing, but then it also decreases. When examining the learning curves in this architecture, we can observe some spikes that are due to the close values in the axis and the lack of activation functions. Based on this analysis, there are two promising models in this architecture. The first model (24, 25, 29, 30) has 64 neurons in the first hidden layer and 128 in the second hidden layer. The second model (28, 29, 35, 36) has 256 neurons in the first hidden layer and 64 in the second hidden layer. While both models show promise, I would choose the second model because the increment in time required for training is significant, but the accuracy is much higher.

3.3.3. Architecture 3.

The third architecture will be consisted of the following:

- InputDimension: 300
- FirstHiddenLayer: [16, 32, 64, 128, 256]

- SecondHiddenLayer: [16, 32, 64, 128, 256]
- ThirdHiddenLayer: [16, 32, 64, 128, 256]
- OutputDimension: 3

In the third neural network architecture, there are 125 possible experiments due to the fact that each of the hidden layers has 5 possible sizes. To ensure that the report is as comprehensive as possible, I will present only a few worth showing experiments, but I will provide an explanation for all of them.

Neural Network with 3 Hidden Layer of sizes 16 16 128

Time Taken:	156.238
Average Loss:	1.0818814945148036
F1-Score:	0.3934070467948913
Accuracy:	0.3933486342430115
Recall:	0.3933486342430115
Precision:	0.3935685157775879

Table 37: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.38	0.38	0.38	12210
NEUTRAL	0.37	0.40	0.38	12210
POSITIVE	0.40	0.38	0.39	12210
accuracy			0.38	36630
macro avg	0.38	0.38	0.38	36630
weighted avg	0.38	0.38	0.38	36630

Table 38: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.40	0.39	0.39	1744
NEUTRAL	0.38	0.39	0.38	1744
POSITIVE	0.41	0.40	0.40	1744
accuracy			0.39	5232
macro avg	0.39	0.39	0.39	5232
weighted avg	0.39	0.39	0.39	5232

Table 39: Val Classification Report

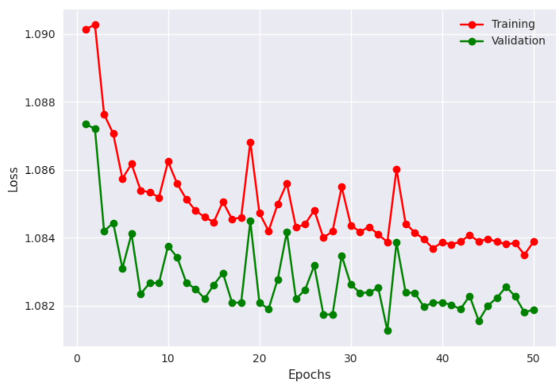

Figure 30: Learnig Curve (Loss)

Figure 31: Learnig Curve (F1-Score)

Neural Network with 3 Hidden Layer of sizes 16 128 16

Time Taken:	158.175
Average Loss:	1.0823867767593547
F1-Score:	0.3899803459644317
Accuracy:	0.3904816508293152
Recall:	0.3904816508293152
Precision:	0.3905250430107116

Table 40: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.39	0.36	0.37	12210
NEUTRAL	0.38	0.48	0.38	12210
POSITIVE	0.39	0.41	0.40	12210
accuracy			0.38	36630
macro avg	0.38	0.38	0.38	36630
weighted avg	0.38	0.38	0.38	36630

Table 41: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.40	0.37	0.39	1744
NEUTRAL	0.37	0.37	0.37	1744
POSITIVE	0.39	0.43	0.41	1744
accuracy			0.39	5232
macro avg	0.39	0.39	0.39	5232
weighted avg	0.39	0.39	0.39	5232

Table 42: Val Classification Report

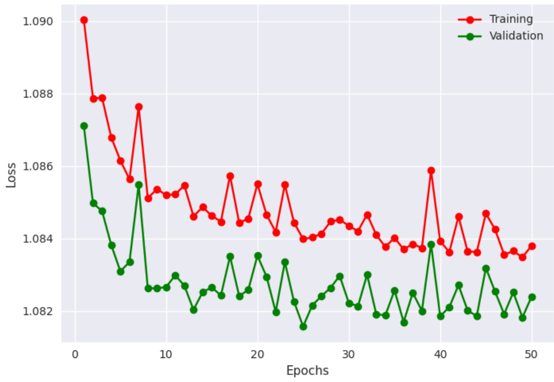

Figure 32: Learnig Curve (Loss)

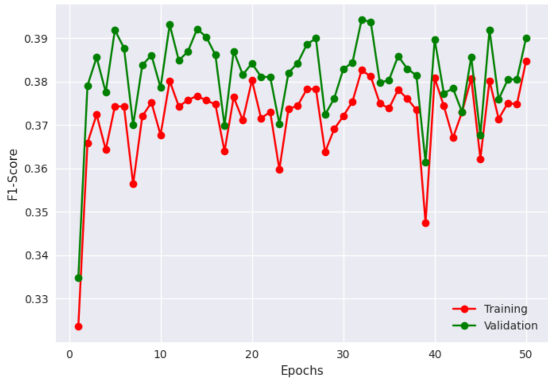

Figure 33: Learnig Curve (F1-Score)

Neural Network with 3 Hidden Layer of sizes 16 256 16

Time Taken:	165.623
Average Loss:	1.0824101350358502
F1-Score:	0.3867201805114746
Accuracy:	0.3876146674156189
Recall:	0.3876146674156189
Precision:	0.3896142244338989

Table 43: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.38	0.41	0.40	12210
NEUTRAL	0.37	0.42	0.40	12210
POSITIVE	0.41	0.32	0.36	12210
accuracy			0.38	36630
macro avg	0.39	0.38	0.38	36630
weighted avg	0.39	0.38	0.38	36630

Table 44: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.39	0.42	0.41	1744
NEUTRAL	0.37	0.41	0.39	1744
POSITIVE	0.41	0.33	0.37	1744
accuracy			0.39	5232
macro avg	0.39	0.39	0.39	5232
weighted avg	0.39	0.39	0.39	5232

Table 45: Val Classification Report

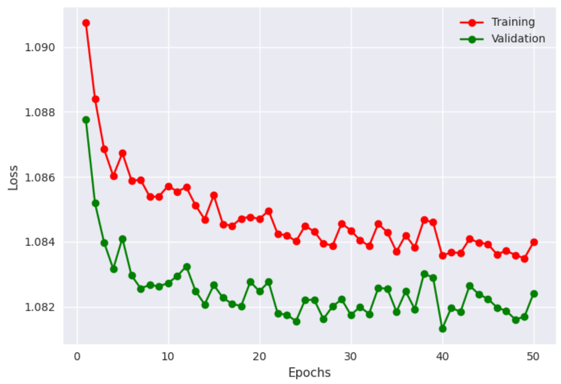

Figure 34: Learnig Curve (Loss)

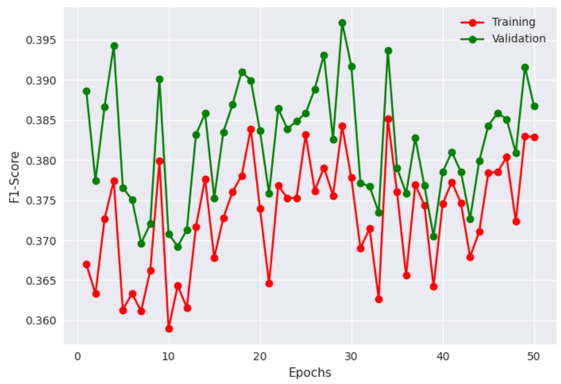

Figure 35: Learnig Curve (F1-Score)

Neural Network with 3 Hidden Layer of sizes 32 32 16

Time Taken:	166.987
Average Loss:	1.0821455753542233
F1-Score:	0.3926104307174682
Accuracy:	0.3925840854644775
Recall:	0.3925840854644775
Precision:	0.3932290077209472

Table 46: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.38	0.38	0.38	12210
NEUTRAL	0.37	0.40	0.39	12210
POSITIVE	0.40	0.37	0.38	12210
accuracy			0.39	36630
macro avg	0.39	0.39	0.39	36630
weighted avg	0.39	0.39	0.39	36630

Table 47: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.40	0.38	0.39	1744
NEUTRAL	0.38	0.41	0.39	1744
POSITIVE	0.41	0.39	0.40	1744
accuracy			0.39	5232
macro avg	0.39	0.39	0.39	5232
weighted avg	0.39	0.39	0.39	5232

Table 48: Val Classification Report

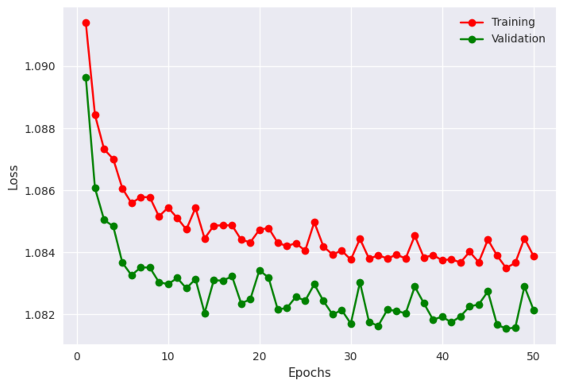

Figure 36: Learnig Curve (Loss)

Figure 37: Learnig Curve (F1-Score)

Neural Network with 3 Hidden Layer of sizes 32 128 64

Time Taken:	175.212
Average Loss:	1.081937353180818
F1-Score:	0.393682926893234
Accuracy:	0.394113153219223
Recall:	0.394113153219223
Precision:	0.393721073865890

Table 49: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.38	0.41	0.39	12210
NEUTRAL	0.38	0.36	0.37	12210
POSITIVE	0.39	0.39	0.39	12210
accuracy			0.39	36630
macro avg	0.39	0.39	0.39	36630
weighted avg	0.39	0.39	0.39	36630

Table 50: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.39	0.41	0.40	1744
NEUTRAL	0.38	0.36	0.37	1744
POSITIVE	0.41	0.42	0.41	1744
accuracy			0.39	5232
macro avg	0.39	0.39	0.39	5232
weighted avg	0.39	0.39	0.39	5232

Table 51: Val Classification Report

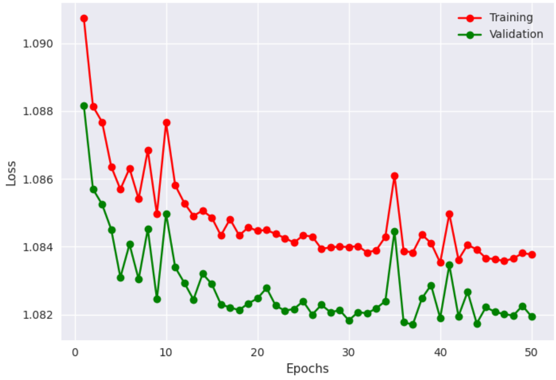

Figure 38: Learnig Curve (Loss)

Figure 39: Learnig Curve (F1-Score)

Neural Network with 3 Hidden Layer of sizes 128 64 256

Time Taken:	206.376
Average Loss:	1.082655940033974
F1-Score:	0.393551617860794
Accuracy:	0.393922001123428
Recall:	0.393922001123428
Precision:	0.395164757966995

Table 52: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.39	0.34	0.36	12210
NEUTRAL	0.37	0.42	0.40	12210
POSITIVE	0.39	0.38	0.39	12210
accuracy			0.38	36630
macro avg	0.38	0.38	0.38	36630
weighted avg	0.38	0.38	0.38	36630

Table 53: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.41	0.35	0.38	1744
NEUTRAL	0.37	0.42	0.39	1744
POSITIVE	0.40	0.41	0.41	1744
accuracy			0.39	5232
macro avg	0.40	0.39	0.39	5232
weighted avg	0.40	0.39	0.39	5232

Table 54: Val Classification Report

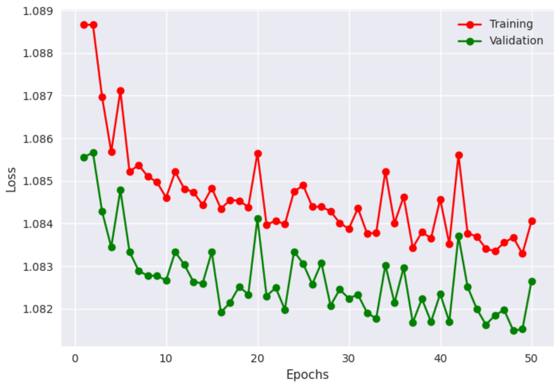

Figure 40: Learnig Curve (Loss)

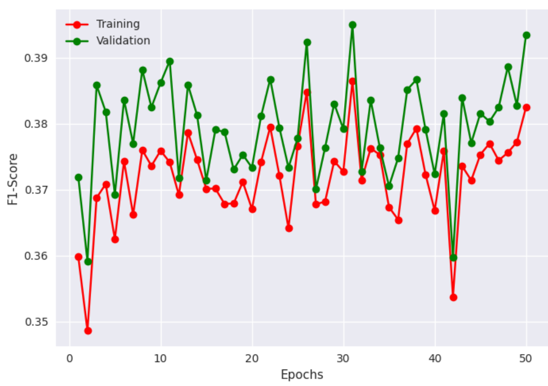

Figure 41: Learnig Curve (F1-Score)

Neural Network with 3 Hidden Layer of sizes 128 128 16

Time Taken:	213.647
Average Loss:	1.081766934205268
F1-Score:	0.396622031927108
Accuracy:	0.396597862243652
Recall:	0.396597862243652
Precision:	0.397094696760177

Table 55: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.38	0.41	0.39	12210
NEUTRAL	0.37	0.39	0.38	12210
POSITIVE	0.40	0.36	0.38	12210
accuracy			0.38	36630
macro avg	0.38	0.38	0.38	36630
weighted avg	0.38	0.38	0.38	36630

Table 56: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.39	0.41	0.40	1744
NEUTRAL	0.38	0.39	0.38	1744
POSITIVE	0.42	0.39	0.40	1744
accuracy			0.40	5232
macro avg	0.40	0.40	0.40	5232
weighted avg	0.40	0.40	0.40	5232

Table 57: Val Classification Report

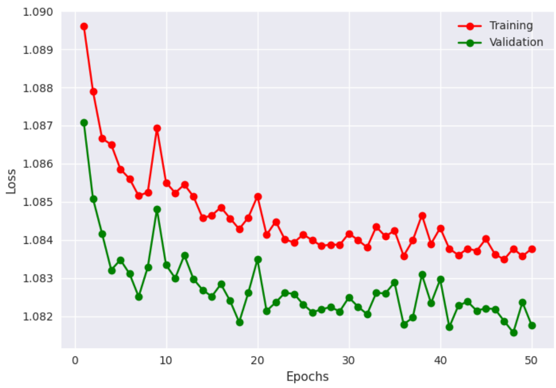

Figure 42: Learnig Curve (Loss)

Figure 43: Learnig Curve (F1-Score)

Neural Network with 3 Hidden Layer of sizes 256 128 16

Time Taken:	204.767
Average Loss:	1.0824205838941288
F1-Score:	0.3873724043369293
Accuracy:	0.3891437351703644
Recall:	0.3891437351703644
Precision:	0.3920158743858337

Table 58: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.38	0.41	0.40	12210
NEUTRAL	0.37	0.44	0.40	12210
POSITIVE	0.41	0.31	0.35	12210
accuracy			0.39	36630
macro avg	0.39	0.39	0.38	36630
weighted avg	0.39	0.39	0.38	36630

Table 59: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.39	0.41	0.40	1744
NEUTRAL	0.37	0.44	0.40	1744
POSITIVE	0.41	0.32	0.36	1744
accuracy			0.39	5232
macro avg	0.39	0.39	0.39	5232
weighted avg	0.39	0.39	0.39	5232

Table 60: Val Classification Report

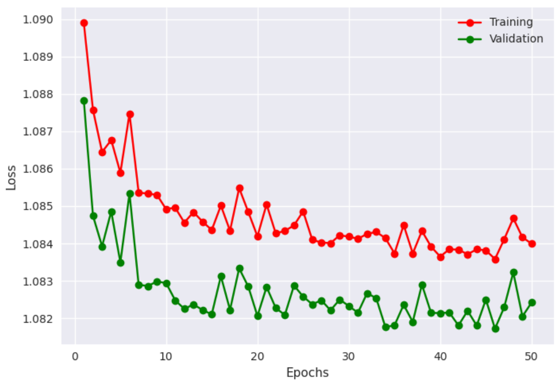

Figure 44: Learnig Curve (Loss)

Figure 45: Learnig Curve (F1-Score)

Analysis of the third architecture In the final analysis of the third architecture, it is evident that as the size of the layers increases, so does the required training time. However, the extended time does not consistently lead to a proportional increase in accuracy. When the first layer has 16 neurons and the second and third layers have more than 16 neurons, the time increases along with the metrics, but when the third layer reaches 256 neurons, the model learns too much, leading to overfitting. Similar challenges arise when the second layer has 128 neurons. When the first layer has 32 neurons, the accuracy decreases with a second layer of 16 neurons, but increasing the second layer to 32 results in a temporary improvement before another decline. The model performs well with 64 neurons in the first layer, offering flexibility in choosing layer sizes. The best results are achieved with 128 neurons in the first and second layers and 16 neurons in the third layer, resulting in a highly accurate model with balanced metrics. The model with 128 neurons in the first two layers and 16 in the third layer shows the most potential, despite the longer training time. For the 128 128 16 model, the classification reports are well-balanced, and two out of the three sentiments have the same F1 score in the classification report. However, the 256 neurons in the first layer result in the same problem of having too many neurons, limiting the model to fewer neurons in the next layers. In conclusion, the model with size 128 128 16 (42, 43, 56, 57) has the most potential, but it may be costly in terms of training time. Further experimentation aims to balance the time and performance for this promising configuration.

3.4. Further Experiments

In our previous experiments, we have focused on "pure" architectures, where we only varied the number of hidden layers and their sizes. Among these architectures, the model from the third architecture (42, 43, 56, 57), with three hidden layers of sizes 128 128 16, has demonstrated the most promising results. The classification report is well-distributed, and the model's performance is satisfactory. Now, we will subject this model to further experimentation, where we will explore various activation functions, optimizers, dropout ratios, and learning rates. By fine-tuning these components, we aim to improve the model's performance and optimize its generalization capabilities. This iterative process will help us identify the optimal combination of hyperparameters for our specific problem, ensuring that our model achieves the best possible results.

3.4.1. Activation Function.

NN with 3 Hidden Layer of sizes 128 128 16 & ReLU

Time Taken:	159.711
Average Loss:	1.1197934907146185
F1-Score:	0.3585146367549896
Accuracy:	0.3753822743892669
Recall:	0.3753822743892669
Precision:	0.3879470527172088

Table 61: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.48	0.43	0.45	12210
NEUTRAL	0.41	0.68	0.51	12210
POSITIVE	0.58	0.26	0.36	12210
accuracy			0.46	36630
macro avg	0.49	0.46	0.44	36630
weighted avg	0.49	0.46	0.44	36630

Table 62: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.39	0.35	0.37	1744
NEUTRAL	0.36	0.58	0.44	1744
POSITIVE	0.42	0.19	0.27	1744
accuracy			0.38	5232
macro avg	0.39	0.38	0.36	5232
weighted avg	0.39	0.38	0.36	5232

Table 63: Val Classification Report

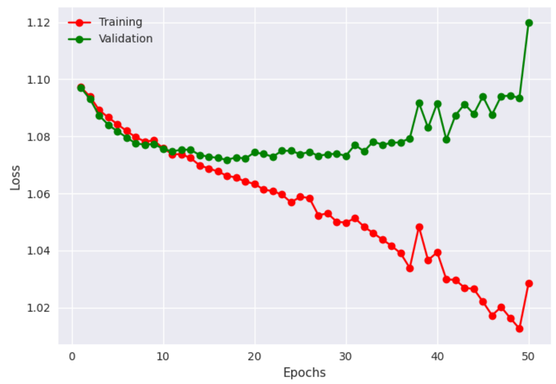

Figure 46: Learnig Curve (Loss)

Figure 47: Learnig Curve (F1-Score)

NN with 3 Hidden Layer of sizes 128 128 16 & CELU

Time Taken:	164.181
Average Loss:	1.0734105677050552
F1-Score:	0.3873679041862488
Accuracy:	0.3916284739971161
Recall:	0.3916284739971161
Precision:	0.3985347151756286

Table 64: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.40	0.40	0.40	12210
NEUTRAL	0.38	0.50	0.43	12210
POSITIVE	0.44	0.29	0.35	12210
accuracy			0.40	36630
macro avg	0.41	0.40	0.40	36630
weighted avg	0.41	0.40	0.40	36630

Table 65: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.40	0.40	0.40	1744
NEUTRAL	0.36	0.49	0.42	1744
POSITIVE	0.43	0.28	0.34	1744
accuracy			0.39	5232
macro avg	0.40	0.39	0.39	5232
weighted avg	0.40	0.39	0.39	5232

Table 66: Val Classification Report

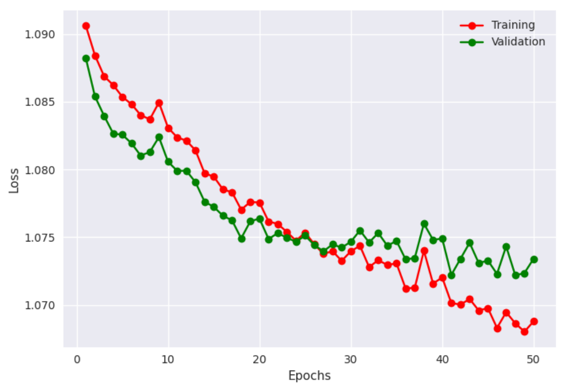

Figure 48: Learnig Curve (Loss)

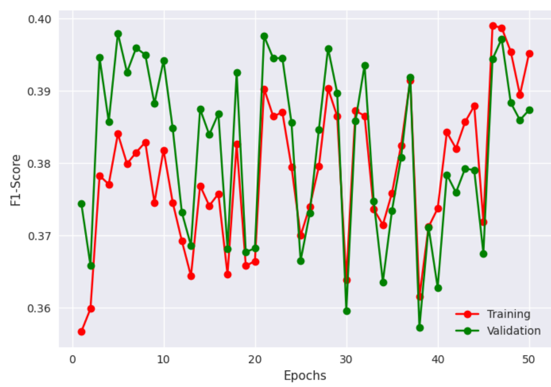

Figure 49: Learnig Curve (F1-Score)

NN with 3 Hidden Layer of sizes 128 128 16 & GELU

Time Taken:	197.831
Average Loss:	1.0700904045265385
F1-Score:	0.4029290080070495
Accuracy:	0.4067278206348419
Recall:	0.4067278206348419
Precision:	0.4076241850852966

Table 67: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.39	0.50	0.44	12210
NEUTRAL	0.41	0.35	0.37	12210
POSITIVE	0.42	0.36	0.39	12210
accuracy			0.40	36630
macro avg	0.41	0.40	0.40	36630
weighted avg	0.41	0.40	0.40	36630

Table 68: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.40	0.52	0.45	1744
NEUTRAL	0.41	0.33	0.37	1744
POSITIVE	0.42	0.37	0.39	1744
accuracy			0.41	5232
macro avg	0.41	0.41	0.40	5232
weighted avg	0.41	0.41	0.40	5232

Table 69: Val Classification Report

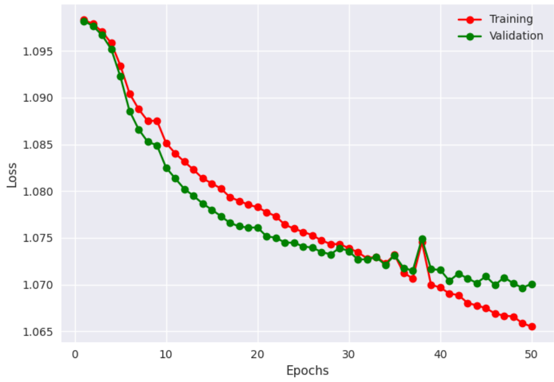

Figure 50: Learnig Curve (Loss)

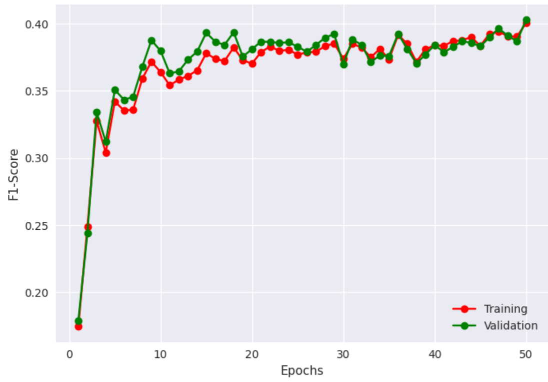

Figure 51: Learnig Curve (F1-Score)

NN with 3 Hidden Layer of sizes 128 128 16 & SiLU

Time Taken:	161.410
Average Loss:	1.0722563108172984
F1-Score:	0.4021279811859131
Accuracy:	0.4063456058502197
Recall:	0.4063456058502197
Precision:	0.4065349698066711

Table 70: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.38	0.49	0.43	12210
NEUTRAL	0.39	0.34	0.36	12210
POSITIVE	0.42	0.36	0.39	12210
accuracy			0.40	36630
macro avg	0.40	0.40	0.39	36630
weighted avg	0.40	0.40	0.39	36630

Table 71: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.40	0.52	0.45	1744
NEUTRAL	0.40	0.32	0.35	1744
POSITIVE	0.42	0.38	0.40	1744
accuracy			0.41	5232
macro avg	0.41	0.41	0.40	5232
weighted avg	0.41	0.41	0.40	5232

Table 72: Val Classification Report

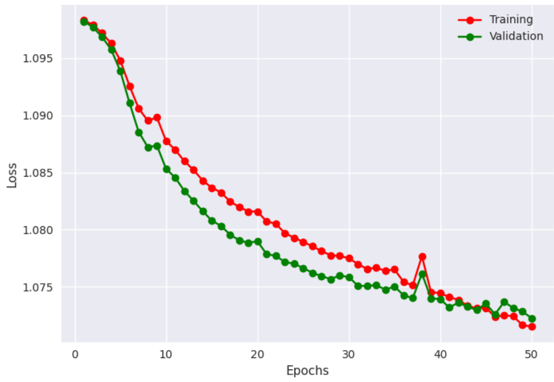

Figure 52: Learnig Curve (Loss)

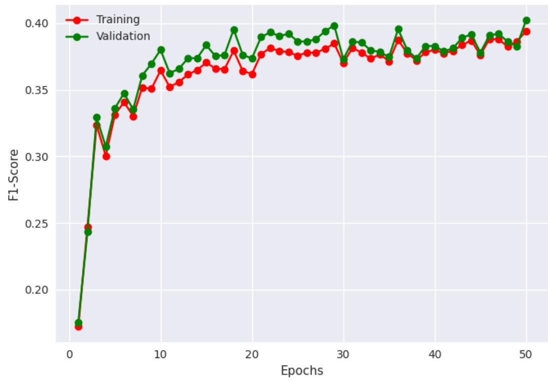

Figure 53: Learnig Curve (F1-Score)

The introduction of activation functions has greatly improved the stability of the curves during our neural network training. The loss curves show a steady and almost linear decline, suggesting enhanced convergence, while the F1 curves display a corresponding linear increase, highlighting the models' capability to detect complex patterns in the data. It is worth noting that the CELU activation function has noticeable spikes in the F1 curve, although the closeness of values along the axis should be considered. Furthermore, with the current number of epochs, most activation functions show signs of overfitting. Given the crucial role activation functions play in model performance, we intend to conduct additional experiments. We will assign an optimal number of epochs to each activation function, based on our initial observations from the loss and F1 curves, to tackle overfitting and gain a deeper understanding of their influence on model generalization and performance.

3.4.2. Activation Function Best Epoch.

—— NN with 3 Hidden Layer of sizes 128 128 16 & ReLU & Epoch 10 ——

Time Taken:	32.469
Average Loss:	1.075634155798396
F1-Score:	0.392484903335571
Accuracy:	0.401949554681777
Recall:	0.401949554681777
Precision:	0.407101511955261

Table 73: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.38	0.56	0.46	12210
NEUTRAL	0.39	0.34	0.36	12210
POSITIVE	0.43	0.28	0.34	12210
accuracy			0.40	36630
macro avg	0.40	0.40	0.39	36630
weighted avg	0.40	0.40	0.39	36630

Table 74: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.39	0.58	0.47	1744
NEUTRAL	0.39	0.33	0.36	1744
POSITIVE	0.44	0.29	0.35	1744
accuracy			0.40	5232
macro avg	0.41	0.40	0.39	5232
weighted avg	0.41	0.40	0.39	5232

Table 75: Val Classification Report

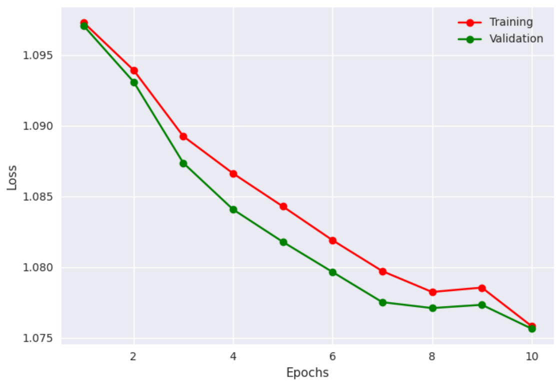

Figure 54: Learnig Curve (Loss)

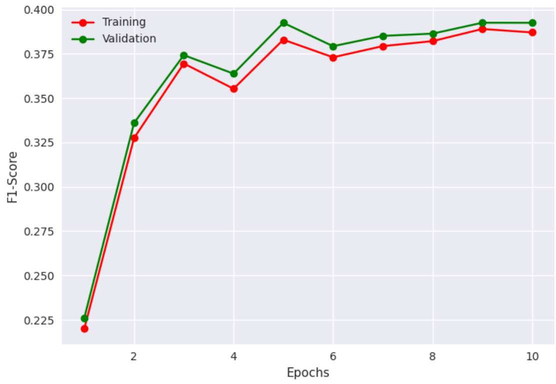

Figure 55: Learnig Curve (F1-Score)

NN with 3 Hidden Layer of sizes 128 128 16 & CELU & Epoch 25

Time Taken:	83.301
Average Loss:	1.0751167240492794
F1-Score:	0.3665035665035248
Accuracy:	0.3971712589263916
Recall:	0.3971712589263916
Precision:	0.4093601405620575

Table 76: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.37	0.70	0.49	12210
NEUTRAL	0.43	0.22	0.29	12210
POSITIVE	0.44	0.27	0.33	12210
accuracy			0.40	36630
macro avg	0.41	0.40	0.37	36630
weighted avg	0.41	0.40	0.37	36630

Table 77: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.38	0.73	0.50	1744
NEUTRAL	0.40	0.20	0.27	1744
POSITIVE	0.45	0.26	0.33	1744
accuracy			0.40	5232
macro avg	0.41	0.40	0.37	5232
weighted avg	0.41	0.40	0.37	5232

Table 78: Val Classification Report

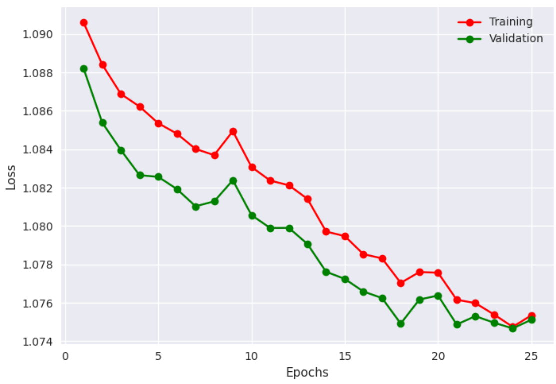

Figure 56: Learnig Curve (Loss)

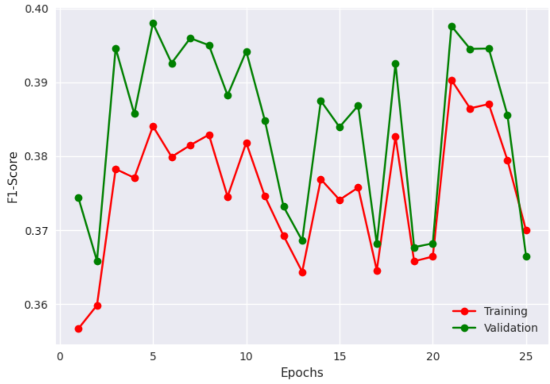

Figure 57: Learnig Curve (F1-Score)

NN with 3 Hidden Layer of sizes 128 128 16 & GELU & Epoch 30

Time Taken:	119.368
Average Loss:	1.073575013821278
F1-Score:	0.369747906923294
Accuracy:	0.396597862243652
Recall:	0.396597862243652
Precision:	0.405326008796691

Table 79: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.37	0.66	0.48	12210
NEUTRAL	0.43	0.20	0.27	12210
POSITIVE	0.43	0.33	0.37	12210
accuracy			0.40	36630
macro avg	0.41	0.40	0.37	36630
weighted avg	0.41	0.40	0.37	36630

Table 80: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.38	0.68	0.49	1744
NEUTRAL	0.42	0.18	0.25	1744
POSITIVE	0.42	0.32	0.36	1744
accuracy			0.40	5232
macro avg	0.41	0.40	0.37	5232
weighted avg	0.41	0.40	0.37	5232

Table 81: Val Classification Report

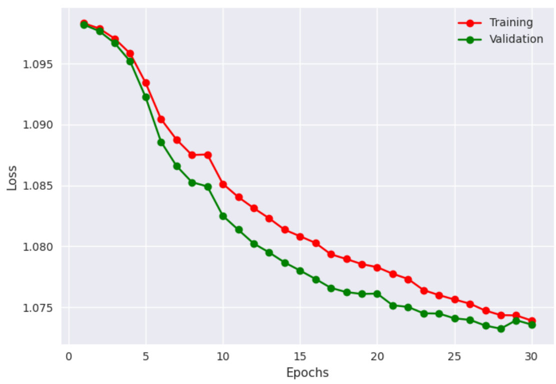

Figure 58: Learnig Curve (Loss)

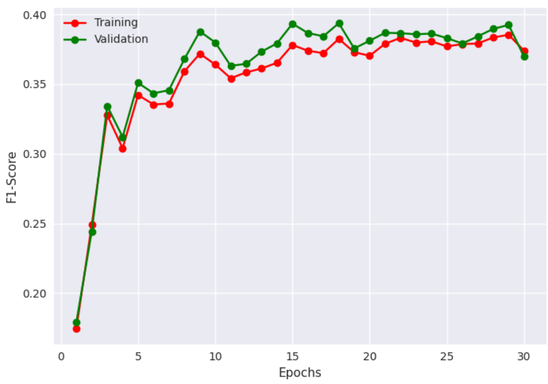

Figure 59: Learnig Curve (F1-Score)

NN with 3 Hidden Layer of sizes 128 128 16 & SiLU & Epoch 45

Time Taken:	145.984
Average Loss:	1.073518342198946
F1-Score:	0.377934694290161
Accuracy:	0.401567280292511
Recall:	0.401567280292511
Precision:	0.41461938619613

Table 82: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.38	0.51	0.44	12210
NEUTRAL	0.46	0.17	0.25	12210
POSITIVE	0.39	0.51	0.44	12210
accuracy			0.40	36630
macro avg	0.41	0.40	0.38	36630
weighted avg	0.41	0.40	0.38	36630

Table 83: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.40	0.52	0.45	1744
NEUTRAL	0.45	0.16	0.23	1744
POSITIVE	0.39	0.53	0.45	1744
accuracy			0.40	5232
macro avg	0.41	0.40	0.38	5232
weighted avg	0.41	0.40	0.38	5232

Table 84: Val Classification Report

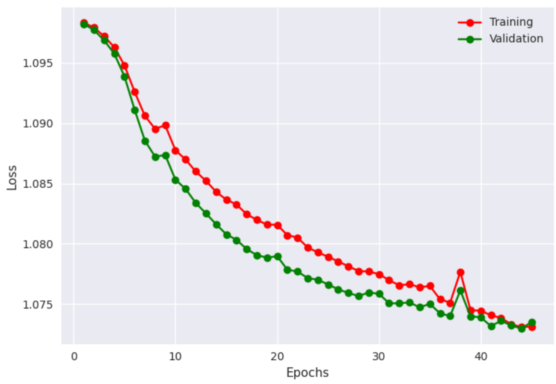

Figure 60: Learnig Curve (Loss)

Figure 61: Learnig Curve (F1-Score)

Once each activation function is trained with its optimal number of epochs, a clearer understanding is gained. The ReLU activation function with 10 epochs excels, offering a reasonable training time and superior metrics compared to the others. The loss and F1 score curves show a desirable linearity, and the classification report indicates a balanced performance on both the training and validation sets. However, the CELU activation function with 25 epochs experiences an increase in training time, accompanied by a substantial decline in the F1 score. The inconsistent distribution of sentiments suggests that it does not meet acceptable error margins. The GELU activation function with 30 epochs displays smooth curves but suffers from extended training times, low metrics, and unreliable sentiment classification. The SiLU activation function with 45 epochs faces similar challenges. In summary, the ReLU activation function with 10 epochs (54, 55, 74, 75) proves to be the most favorable choice, as it effectively balances training time and accuracy.

3.4.3. Optimizer Testing.

—— NN with 3 Hidden Layer of sizes 128 128 16 & ReLU & Epoch 10 & SGD ——

Time Taken:	32.847
Average Loss:	1.075634155798396
F1-Score:	0.392484903335571
Accuracy:	0.401949554681777
Recall:	0.401949554681777
Precision:	0.407101511955261

Table 85: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.38	0.56	0.46	12210
NEUTRAL	0.39	0.34	0.36	12210
POSITIVE	0.43	0.28	0.34	12210
accuracy			0.40	36630
macro avg	0.40	0.40	0.39	36630
weighted avg	0.40	0.40	0.39	36630

Table 86: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.39	0.58	0.47	1744
NEUTRAL	0.39	0.33	0.36	1744
POSITIVE	0.44	0.29	0.35	1744
accuracy			0.40	5232
macro avg	0.41	0.40	0.39	5232
weighted avg	0.41	0.40	0.39	5232

Table 87: Val Classification Report

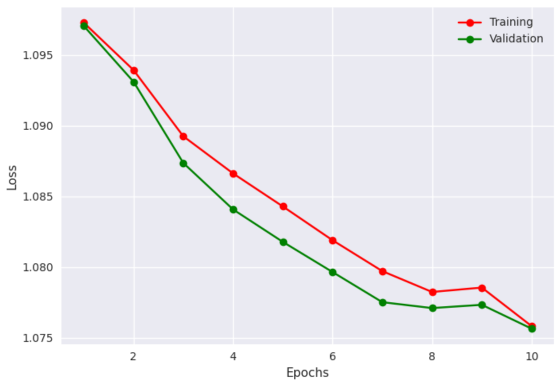

Figure 62: Learnig Curve (Loss)

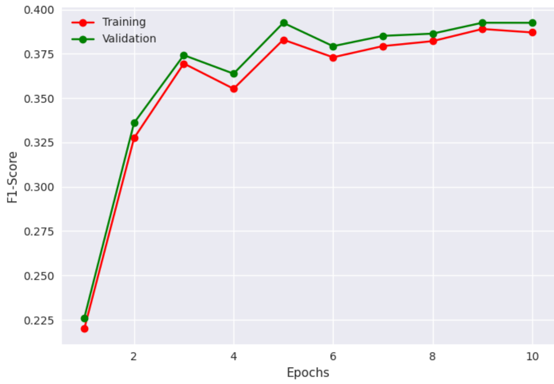

Figure 63: Learnig Curve (F1-Score)

NN with 3 Hidden Layer of sizes 128 128 16 & ReLU & Epoch 10 & Adagrad

Time Taken:	38.673
Average Loss:	1.0713630767043578
F1-Score:	0.3826901614665985
Accuracy:	0.3914373219013214
Recall:	0.3914373219013214
Precision:	0.3967621922492981

Table 88: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.40	0.61	0.48	12210
NEUTRAL	0.45	0.36	0.40	12210
POSITIVE	0.46	0.32	0.38	12210
accuracy			0.43	36630
macro avg	0.44	0.43	0.42	36630
weighted avg	0.44	0.43	0.42	36630

Table 89: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.38	0.57	0.45	1744
NEUTRAL	0.39	0.31	0.35	1744
POSITIVE	0.42	0.30	0.35	1744
accuracy			0.39	5232
macro avg	0.40	0.39	0.38	5232
weighted avg	0.40	0.39	0.38	5232

Table 90: Val Classification Report

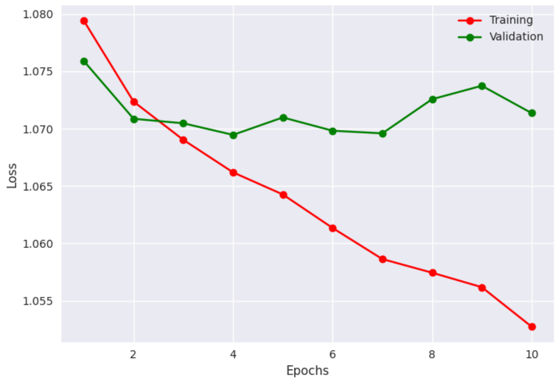

Figure 64: Learnig Curve (Loss)

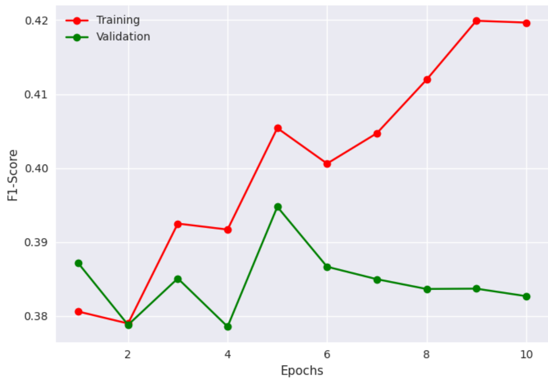

Figure 65: Learnig Curve (F1-Score)

NN with 3 Hidden Layer of sizes 128 128 16 & ReLU & Epoch 10 & Adamax

Time Taken:	57.432
Average Loss:	1.0757253277556977
F1-Score:	0.3838359117507934
Accuracy:	0.3931574821472168
Recall:	0.3931574821472168
Precision:	0.4065655767917633

Table 91: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.40	0.61	0.48	12210
NEUTRAL	0.45	0.36	0.40	12210
POSITIVE	0.46	0.32	0.38	12210
accuracy			0.43	36630
macro avg	0.44	0.43	0.42	36630
weighted avg	0.44	0.43	0.42	36630

Table 92: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.38	0.57	0.45	1744
NEUTRAL	0.39	0.31	0.35	1744
POSITIVE	0.42	0.30	0.35	1744
accuracy			0.39	5232
macro avg	0.40	0.39	0.38	5232
weighted avg	0.40	0.39	0.38	5232

Table 93: Val Classification Report

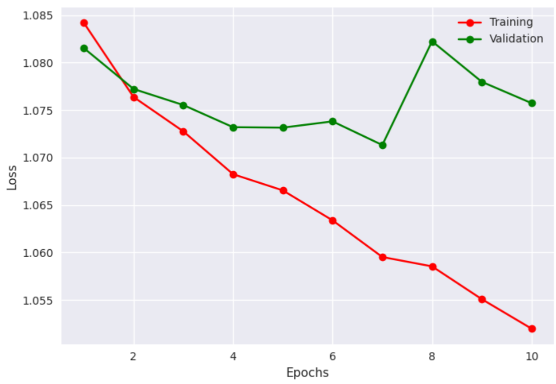

Figure 66: Learnig Curve (Loss)

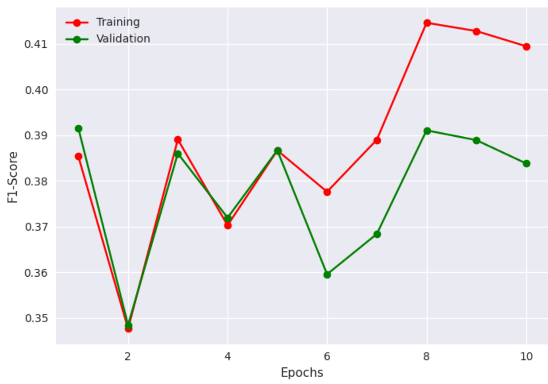

Figure 67: Learnig Curve (F1-Score)

I initiated our tests using the SGD optimizer and observed that the training and validation curves are almost identical, indicating a well-balanced model that is neither overfitting nor underfitting. The classification reports for both the train and validation sets also showed similar results. However, I noticed a slight bias towards the negative sentiment, which I was unable to eliminate later on. Next, I experimented with the Adagrad optimizer and found that the model converged at only 2 epochs. Although this may seem advantageous in terms of training time, I am concerned about the potential for overfitting. Rapid convergence can lead to a model that is too specific to the training data, resulting in poor generalization to new, unseen data. Similarly, the Adamax optimizer also resulted in quick convergence, making it unsuitable for our purposes. Therefore, I have decided to proceed with the model that utilizes the SGD optimizer (66, 67, 92, 93), as it provides the most balanced and reliable results for our specific use case.

3.4.4. Dropout Testing.

NN with 3 Hidden Layer of sizes 128 128 16 & ReLU & Epoch 10 & SGD & Dropout 0.01

Time Taken:	36.808
Average Loss:	1.07492910339198
F1-Score:	0.39419555664062
Accuracy:	0.40252292156219
Recall:	0.40252292156219
Precision:	0.40402811765670

Table 94: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.39	0.52	0.44	12210
NEUTRAL	0.41	0.27	0.33	12210
POSITIVE	0.41	0.41	0.41	12210
accuracy			0.40	36630
macro avg	0.40	0.40	0.39	36630
weighted avg	0.40	0.40	0.39	36630

Table 95: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.40	0.53	0.46	1744
NEUTRAL	0.41	0.26	0.32	1744
POSITIVE	0.40	0.42	0.41	1744
accuracy			0.40	5232
macro avg	0.40	0.40	0.39	5232
weighted avg	0.40	0.40	0.39	5232

Table 96: Val Classification Report

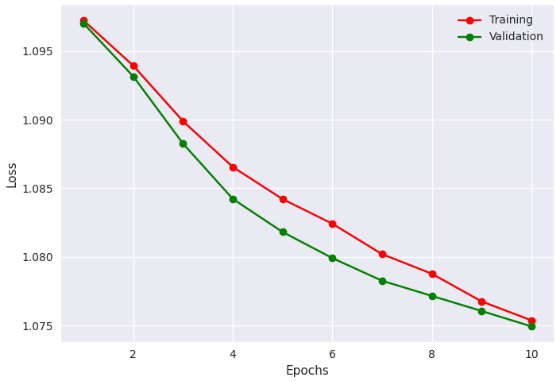

Figure 68: Learnig Curve (Loss)

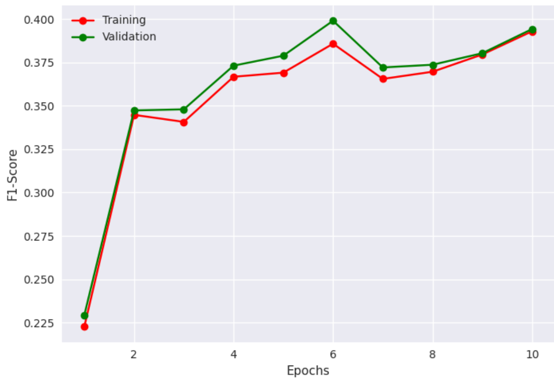

Figure 69: Learnig Curve (F1-Score)

NN with 3 Hidden Layer of sizes 128 128 16 & ReLU & Epoch 10 & SGD & Dropout 0.02

Time Taken:	36.383
Average Loss:	1.0750369281943786
F1-Score:	0.3952924013137817
Accuracy:	0.4044342637062073
Recall:	0.4044342637062073
Precision:	0.4059419631958008

Table 97: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.39	0.52	0.44	12210
NEUTRAL	0.41	0.27	0.32	12210
POSITIVE	0.41	0.41	0.41	12210
accuracy			0.40	36630
macro avg	0.40	0.40	0.39	36630
weighted avg	0.40	0.40	0.39	36630

Table 98: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.40	0.54	0.46	1744
NEUTRAL	0.41	0.25	0.31	1744
POSITIVE	0.41	0.42	0.41	1744
accuracy			0.40	5232
macro avg	0.41	0.40	0.40	5232
weighted avg	0.41	0.40	0.40	5232

Table 99: Val Classification Report

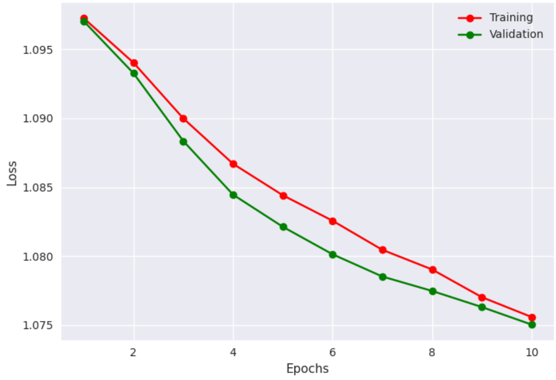

Figure 70: Learnig Curve (Loss)

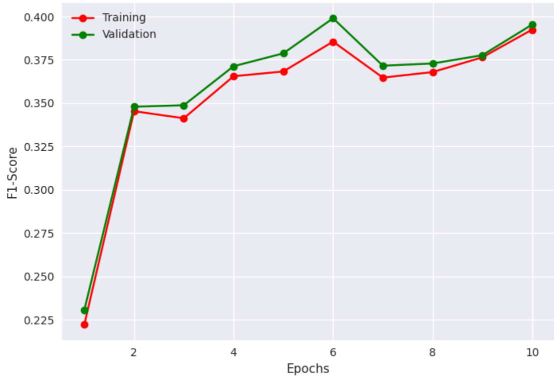

Figure 71: Learnig Curve (F1-Score)

NN with 3 Hidden Layer of sizes 128 128 16 & ReLU & Epoch 10 & SGD & Dropout 0.03

Time Taken:	36.992
Average Loss:	1.0752167860302357
F1-Score:	0.3963801860809326
Accuracy:	0.4051987826824188
Recall:	0.4051987826824188
Precision:	0.4067874252796173

Table 100: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.39	0.52	0.44	12210
NEUTRAL	0.41	0.27	0.33	12210
POSITIVE	0.41	0.41	0.41	12210
accuracy			0.40	36630
macro avg	0.40	0.40	0.39	36630
weighted avg	0.40	0.40	0.39	36630

Table 101: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.40	0.55	0.46	1744
NEUTRAL	0.41	0.26	0.32	1744
POSITIVE	0.41	0.41	0.41	1744
accuracy			0.41	5232
macro avg	0.41	0.41	0.40	5232
weighted avg	0.41	0.41	0.40	5232

Table 102: Val Classification Report

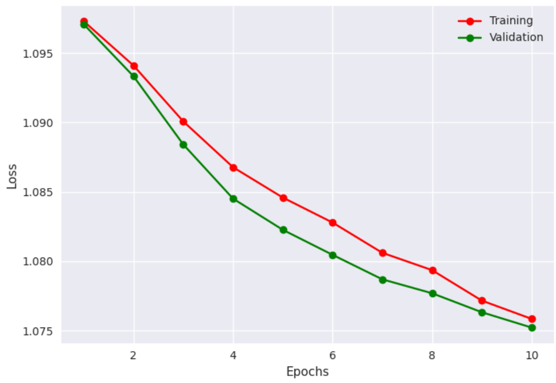

Figure 72: Learnig Curve (Loss)

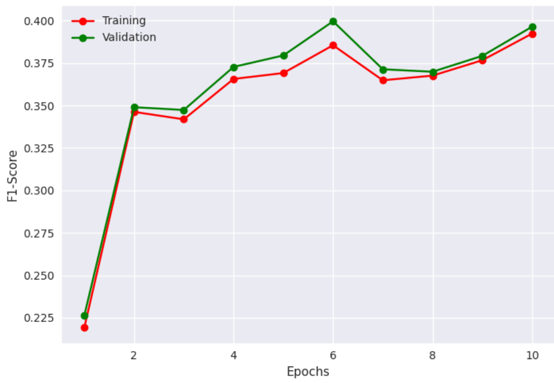

Figure 73: Learnig Curve (F1-Score)

The use of dropout in our model has resulted in a well-balanced performance, as evidenced by the fact that the model converges more quickly when using dropout ratios.

However, there is a noticeable lack of improvement in accuracy, which is disappointing considering the additional complexity and training time required. While dropout helps prevent overfitting and encourages the network to learn a more sparse representation, it seems that the benefits of dropout may not outweigh the drawbacks in this particular case given the fact that our model is well-balanced without the dropout. I will continue my experiments without the addition of the dropout layers.

3.4.5. Learning Rate.

NN with 3 Hidden Layer of sizes 128 128 16 & ReLU & Epoch 10 & SGD & Learnin Rate 0.005

Time Taken:	32.859
Average Loss:	1.081512274545267
F1-Score:	0.388637006282806
Accuracy:	0.401758402585983
Recall:	0.401758402585983
Precision:	0.406982600688934

Table 103: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.38	0.59	0.46	12210
NEUTRAL	0.40	0.26	0.32	12210
POSITIVE	0.42	0.32	0.36	12210
accuracy			0.39	36630
macro avg	0.40	0.39	0.38	36630
weighted avg	0.40	0.39	0.38	36630

Table 104: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.39	0.61	0.48	1744
NEUTRAL	0.41	0.26	0.32	1744
POSITIVE	0.42	0.33	0.37	1744
accuracy			0.40	5232
macro avg	0.41	0.40	0.39	5232
weighted avg	0.41	0.40	0.39	5232

Table 105: Val Classification Report

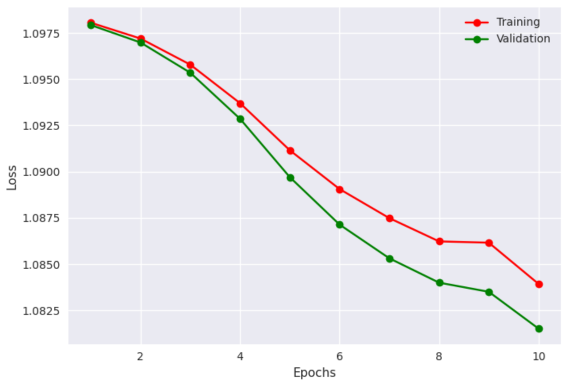

Figure 74: Learnig Curve (Loss)

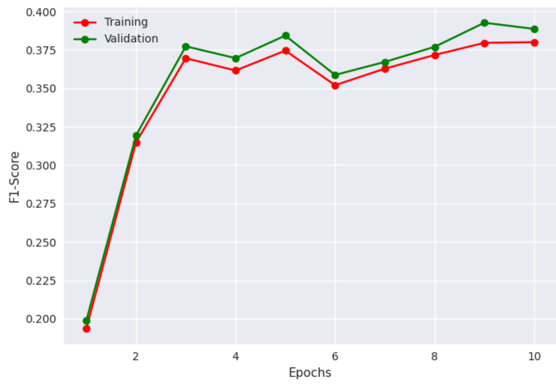

Figure 75: Learnig Curve (F1-Score)

NN with 3 Hidden Layer of sizes 128 128 16 & ReLU & Epoch 10 & SGD & Learnin Rate 0.02

Time Taken:	32.561
Average Loss:	1.07339081122605
F1-Score:	0.38805279135704
Accuracy:	0.39449542760849
Recall:	0.39449542760849
Precision:	0.39889413118362

Table 106: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.39	0.51	0.45	12210
NEUTRAL	0.39	0.42	0.41	12210
POSITIVE	0.45	0.28	0.35	12210
accuracy			0.41	36630
macro avg	0.41	0.41	0.40	36630
weighted avg	0.41	0.41	0.40	36630

Table 107: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.40	0.52	0.45	1744
NEUTRAL	0.37	0.39	0.38	1744
POSITIVE	0.43	0.27	0.33	1744
accuracy			0.39	5232
macro avg	0.40	0.39	0.39	5232
weighted avg	0.40	0.39	0.39	5232

Table 108: Val Classification Report

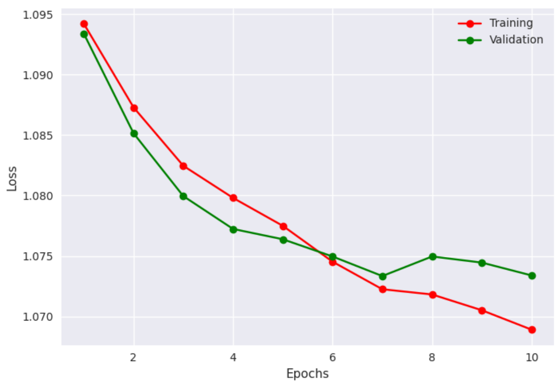

Figure 76: Learnig Curve (Loss)

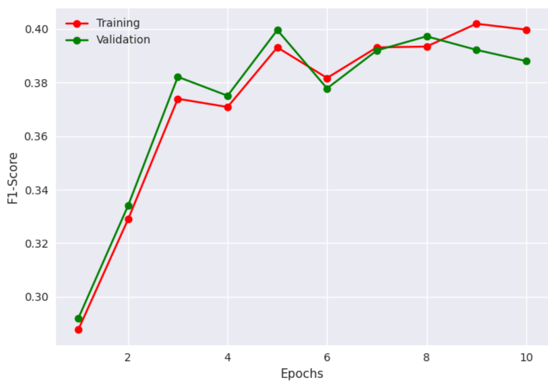

Figure 77: Learnig Curve (F1-Score)

NN with 3 Hidden Layer of sizes 128 128 16 & ReLU & Epoch 10 & SGD & Learnin Rate 0.025

Time Taken:	32.421
Average Loss:	1.072586542604895
F1-Score:	0.388001561164855
Accuracy:	0.391437292098999
Recall:	0.391437292098999
Precision:	0.397402942180633

Table 109: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.40	0.45	0.43	12210
NEUTRAL	0.39	0.47	0.43	12210
POSITIVE	0.45	0.30	0.36	12210
accuracy			0.41	36630
macro avg	0.42	0.41	0.41	36630
weighted avg	0.42	0.41	0.41	36630

Table 110: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.40	0.45	0.42	1744
NEUTRAL	0.36	0.44	0.40	1744
POSITIVE	0.43	0.29	0.34	1744
accuracy			0.39	5232
macro avg	0.40	0.39	0.39	5232
weighted avg	0.40	0.39	0.39	5232

Table 111: Val Classification Report

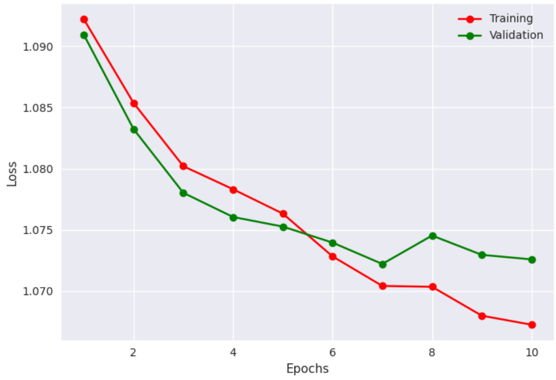

Figure 78: Learnig Curve (Loss)

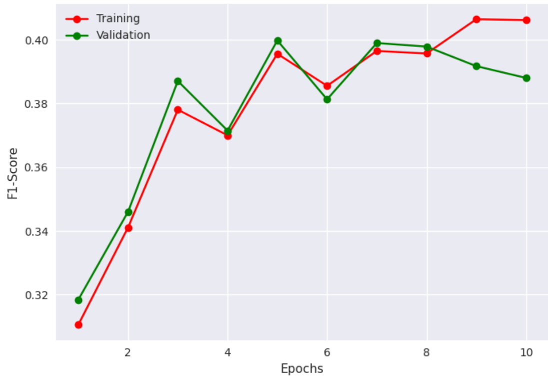

Figure 79: Learnig Curve (F1-Score)

NN with 3 Hidden Layer of sizes 128 128 16 & ReLU & Epoch 10 & SGD & Learnin Rate 0.03

Time Taken:	32.306
Average Loss:	1.0720042892552297
F1-Score:	0.3832414448261261
Accuracy:	0.3912461698055267
Recall:	0.3912461698055267
Precision:	0.4035797715187073

Table 112: Metrics

Training classification report	precision	recall	f1-score	support
NEGATIVE	0.40	0.42	0.41	12210
NEUTRAL	0.38	0.54	0.45	12210
POSITIVE	0.47	0.26	0.34	12210
accuracy			0.41	36630
macro avg	0.42	0.41	0.40	36630
weighted avg	0.42	0.41	0.40	36630

Table 113: Train Classification Report

Validation classification report	precision	recall	f1-score	support
NEGATIVE	0.40	0.41	0.40	1744
NEUTRAL	0.36	0.52	0.43	1744
POSITIVE	0.45	0.25	0.32	1744
accuracy			0.39	5232
macro avg	0.40	0.39	0.38	5232
weighted avg	0.40	0.39	0.38	5232

Table 114: Val Classification Report

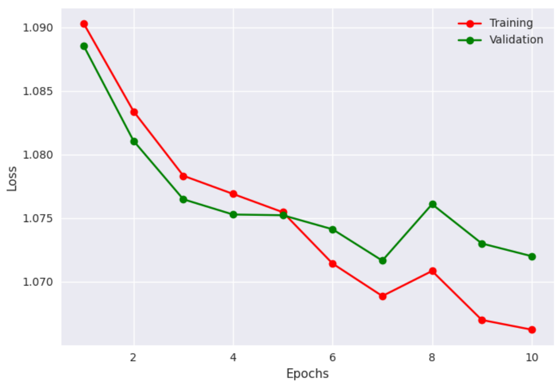

Figure 80: Learnig Curve (Loss)

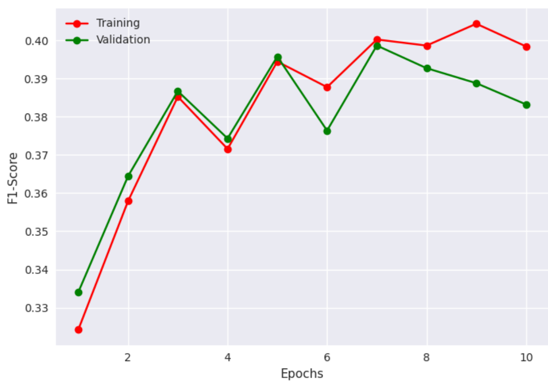

Figure 81: Learnig Curve (F1-Score)

In our current model, we have established a baseline learning rate of 0.01. We have now experimented with various learning rates, including 0.005, 0.02, 0.025, and

0.03. As expected, a smaller learning rate (0.005) results in a longer training time, as the model requires more epochs to converge. This is evident from the widening gap between the loss and f1 curves. On the other hand, learning rates of 0.02, 0.025, and 0.03 lead to overfitting, as the model requires fewer epochs for training. Although we could try reducing the number of epochs to see if the model's performance improves, the high spikes in the f1 curves suggest that the model is unstable and unlikely to stabilize with fewer epochs. Given these observations, we have decided to maintain the learning rate at 0.01 for our final model. This learning rate provides a good balance between training time and model performance, and it has demonstrated stable and consistent results throughout our experiments.

3.4.6. Batch Size. In the course of my assignment, I also conducted experiments using different batch size values. However, I regret to inform that I was unable to save the results of these trials. Nevertheless, I can still draw some conclusions from my observations. As I increased the batch size, I noticed a corresponding decrease in training time. This suggests that our model required less time to complete training with larger batch sizes. Initially, the accuracy of the model remained relatively stable, but this trend did not persist. After a certain batch size (>16) threshold, I observed a decline in accuracy, indicating that the model was having difficulty generalizing. Furthermore, I noted that the losses were higher compared to those presented earlier. These findings suggest that there is an optimal batch size for our model, and that using batch sizes that are too large may negatively impact its performance.

4. Results and Overall Analysis

4.1. Final Model – Logistic Regression

The final Logistic Regression model consists of the following:

- TF-IDF Vectorizer
- C = 0.001
- solver = liblinear
- penalty = l2

Performance

- Time Taken: 3.70
- F1-Score: 0.3778712561197638
- Accuracy: 0.3843646875987297
- Recall: 0.3844635466411971
- Precision: 0.3887267312255672

Below are the represenations of the ROC curve, learning curve and confusion matrix for the final mode:

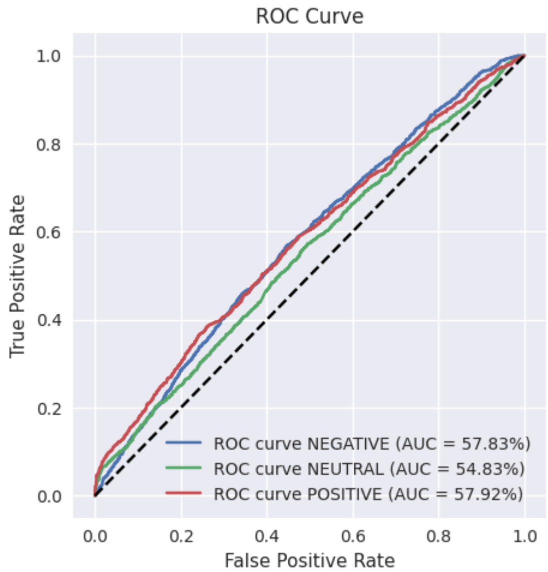

Figure 82: Final Model LR Roc Curve

In Class 1 (NEGATIVE), the AUC is more than 50%, which indicates that the ROC curve is comparatively superior than random chance. That isn't very high, though, suggesting that the model can only slightly outperform chance in distinguishing between instances of the NEGATIVE class. The AUC for Class 2 (NEUTRAL) is 54.83%, which is somewhat greater than 50%. This implies that although there is not much discrimination, the model performs marginally better for the NEUTRAL class than random chance. Class 3 (POSITIVE): The model can reasonably discriminate between cases for the POSITIVE class, according to the AUC of 57.92%. It's not a very strong performance, but it's better than chance.

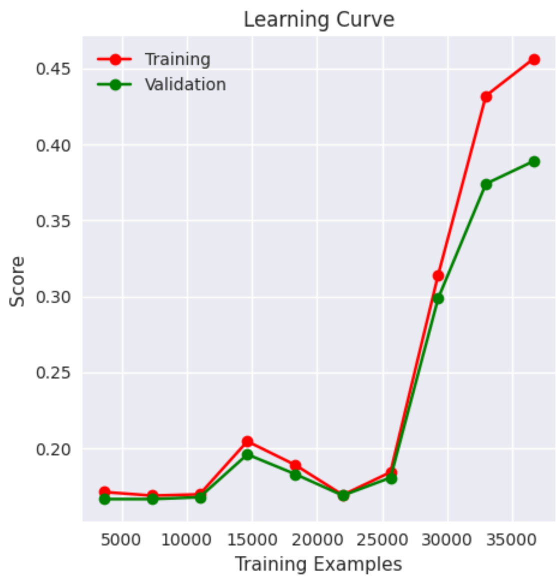

Figure 83: Final Model LR Learning Curve

The learning curve started off slightly, which was evidence of the early lack of skill in the training and validation datasets. But as the iterations went on, there was a noticeable increase. The training curve showed a consistent upward trend, indicating that the model was able to identify more intricate patterns in the data. Simultaneously, the validation curve had an upward trend, indicating the model's excellent applicability to previously unreported data.

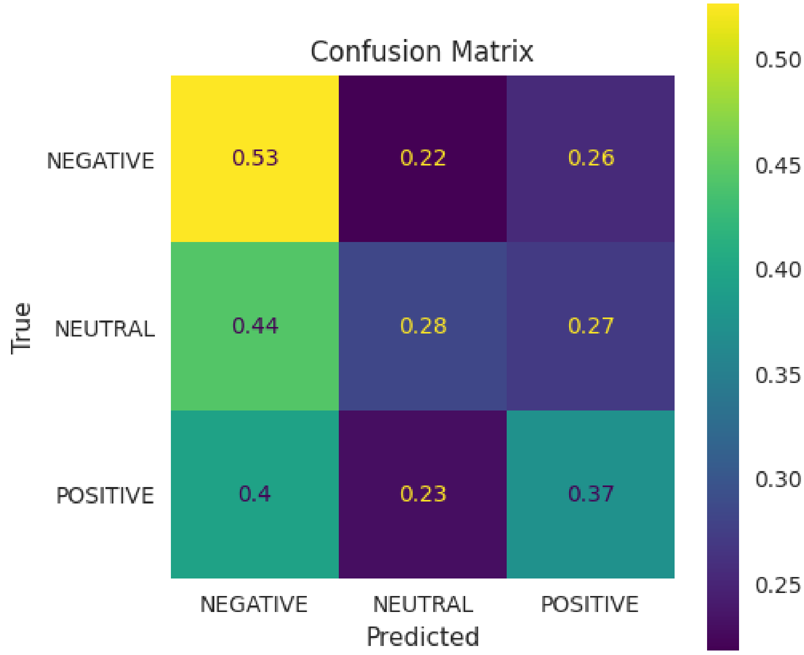

Figure 84: Final Model LR Confusion Matrix

A confusion matrix consists of the following:

- The true NEGATIVE is represented by the first row.
- The true NEUTRAL is represented by the second row.
- The true POSITIVE is represented by the third row.
- The predicted NEGATIVE is represented by the first column.
- The predicted NEUTRAL is represented by the second column.
- The predicted POSITIVE is represented by the third column.

The diagonal values are representing the correct prediction for each class:

- 53% for NEGATIVE
- 28% for NEUTRAL
- 37% for POSITIVE

4.2. Final Model – Feed Forward Neural Network

The final Feed Forward Neural Network model consists of the following:

- Word Embedding
- 3 Hidden Layers
  - 1st hidden layer has size 128
  - 2nd hidden layer has size 128
  - 3rd hidden layer has size 16
- Activation Function = ReLU
- Epochs = 10
- Optimizer = SGD
- Learning Rate = 0.01

Performance

- Time Taken: 40.929
- Average Loss: 1.075634155798396
- F1-Score: 0.392484903335571
- Accuracy: 0.401949554681777
- Recall: 0.401949554681777
- Precision: 0.407101511955261

Below are the represenations of the ROC curve, learning curve and confusion matrix for the final mode:

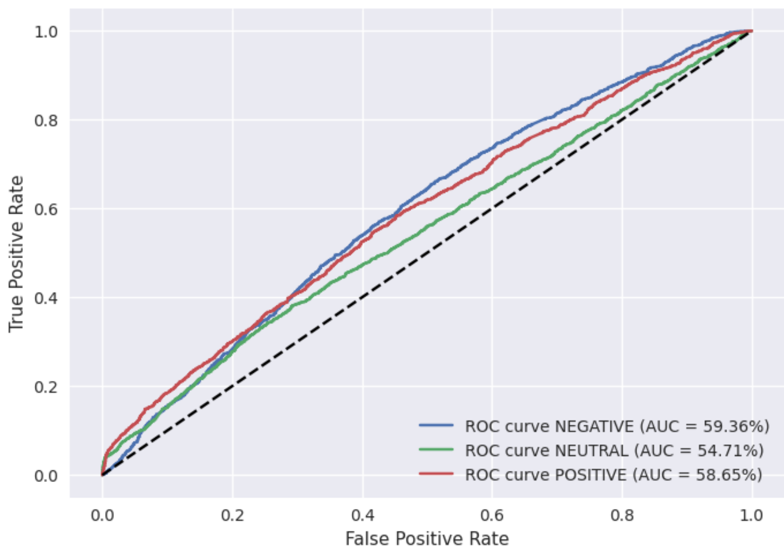

Figure 85: Final Model Roc Curve

The negative class has an AUC (Area Under the Curve) of 59.36%. This means that the classifier performs only slightly better than random guessing when distinguishing between negative samples and the other two classes. The ROC curve for the negative class is relatively flat, indicating that the TPR does not increase much as the FPR increases. The neutral class has an AUC of 54.71%. This is worse than random guessing, indicating that the classifier struggles to distinguish between neutral samples and the other two classes. The ROC curve for the neutral class is also relatively flat, but it is shifted downwards compared to the negative class curve. The positive class has an AUC of 58.65%. This is slightly better than random guessing, but still not a very strong performance. The ROC curve for the positive class is slightly curved upwards, indicating that the TPR increases more rapidly than the FPR as the threshold is decreased. Overall, the ROC curves for all three classes suggest that the classifier is not performing very well. The AUC values are all relatively low, and the ROC curves do not show much separation between the TPR and FPR. This suggests that there may be room for improvement in the classifier's design or training process.

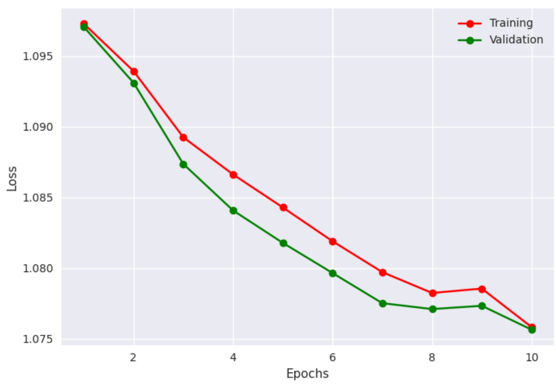

Figure 86: Learnig Curve (Loss)

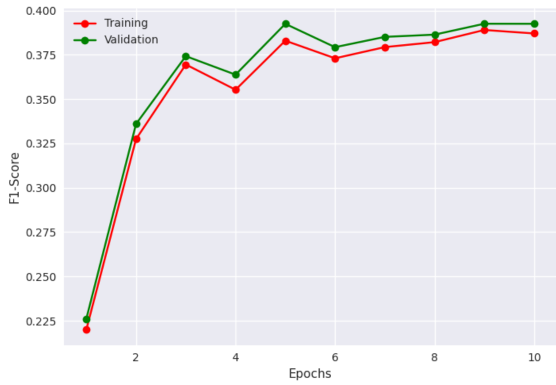

Figure 87: Learnig Curve (F1-Score)

Based on the learning curves, it appears that the model is improving as the number of epochs increases, with both the training and validation loss decreasing and the F1-score increasing for both train and validation sets. The learning curve for the loss shows that as the number of epochs increases, both the training and validation loss decrease, indicating that the model is learning to fit the data better over time. The fact that the validation loss is also decreasing suggests that the model is not overfitting to the training data, which is a good sign. Similarly, the learning curve for the F1-score shows that it is increasing for both the training and validation sets as the number of epochs increases. An increasing F1-score indicates that the model is becoming more accurate in its predictions, and the fact that this is true for both the training and validation sets suggests that the model is not overfitting. Overall, the learning curves suggest that the model is improving over time and is not overfitting to the training data. This is a positive sign and indicates that the model is well-designed and properly tuned.

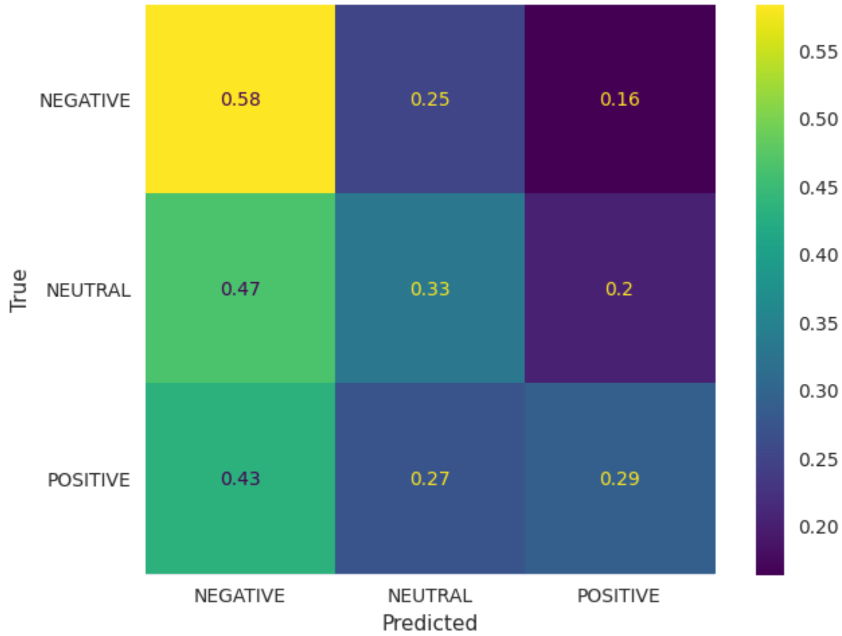

Figure 88: Final Model Confusion Matrix

A confusion matrix consists of the following:

- The true NEGATIVE is represented by the first row.
- The true NEUTRAL is represented by the second row.
- The true POSITIVE is represented by the third row.
- The predicted NEGATIVE is represented by the first column.
- The predicted NEUTRAL is represented by the second column.
- The predicted POSITIVE is represented by the third column.

The diagonal values are representing the correct prediction for each class:

- 58% for NEGATIVE
- 33% for NEUTRAL
- 29% for POSITIVE

4.3. Comparison with the first project

After evaluating the performance of the models in both projects, I have found that while there have been some advancements in the metrics, they are not particularly remarkable. The first project's model has shown to be faster, easier to develop and tune, and provides more flexibility with less effort compared to the second project's model. Although the second project's model may have slightly better performance metrics, the amount of experimentation required to find the optimal model for the given problem makes it a less favorable choice in my opinion. The first project's model is a more practical and efficient solution, making it my preferred choice for this particular task.

5. Conclusion

Metrics show that the model is not as efficient as it could be; metrics are slightly above 40%. Despite that, the extensiveness of the procedure used lays a strong basis for further advancements. The constraints that have been found and the opportunities that remain untapped offer significant insights and may lead to improvements in future editions (next homeworks). This is very important as we go on to the next homework assignment, which is about enhancing our model by investigating different approaches like BERT and Recurrent Neural Networks (RNNs). These varied methods represent a critical step in enhancing the model's functionality and performance by offering a more creative and innovative environment.

6. Bibliography

References

[1] All TorchMetrics — PyTorch-Metrics 1.2.1 documentation.

[2] enable_grad — PyTorch 2.1 documentation.

[3] Feedforward Neural Networks (FNN) - Deep Learning Wizard.

[4] Gensim: topic modelling for humans.

[5] Gensim: topic modelling for humans.

[6] Gensim: topic modelling for humans.

[7] How to Save a Plot to a File Using Matplotlib.

[8] Neural Networks — PyTorch Tutorials 2.2.0+cu121 documentation.

[9] PyTorch Tutorial: Building a Simple Neural Network From Scratch.

[10] sklearn.metrics.classification_report.

[11] torch.nn — PyTorch 2.1 documentation.

[12] torch.optim — PyTorch 2.1 documentation.

[13] Understanding Feed Forward Neural Networks in Deep Learning.

[14] Create a directory in Python, November 2019. Section: Python.

[15] Jason Brownlee. How to use Learning Curves to Diagnose Machine Learning Model Performance, February 2019.

[16] Adrian Rosebrock. Intro to PyTorch: Training your first neural network using PyTorch, July 2021.

[17] Adrian Tam. Develop Your First Neural Network with PyTorch, Step by Step, January 2023.