Layers/Methods	Hyperparameters
Embedding Layer(English dataset)	(input size=14776, output size= 8, input_length=100)
Embedding Layer(Arabic dataset)	(input size=55405, output size= 8, input_length=30)
Convolution Layer (C1)	(Number of filters =128, Kernal size =5)
Activation Function	ReLU Function
Global Max Pooling Layer (P1)	
Dense(Den1)	(units = 10)
Activation Function	ReLU Function
Dense(Den2)	(units = 3)
Activation Function	Sigmoid function

Table V.7 Text CNN Layers/Methods and their hyperparameters

5.4 Conclusion

To conclude, this chapter explained the architecture of the most common deep learning approaches which are CNN and LSTM. We started by talking about the setup environment for the experiments and the necessary libraries, tools, and configurations. Finally, a detailed explanation of how the model was implemented and the selections criteria for the model parameters.

Chapter VI : Results and Discussions

6.1 Introduction

Our idea's major objective is to identify people's emotions through speech and text analysis. For this, we used four CNN models with various architectures, two for audio, two for text, and each two for a different language Arabic and English.

Recent research has shown the ability of convolutional neural networks (CNN) to deal with complex machine problems. Unprecedented results were achieved in tasks such as classification, segmentation, and

object detection, often outperforming human accuracy. CNNs have the ability of learning a hierarchical representation of the input data without requiring any effort to design handcrafted features. Different layers of the network are capable of different levels of abstraction and capture different amount of structure from the patterns present in the data. Due to the complexity of the tasks and the very large number of network parameters that need to be learned during training, CNNs require a massive amount of annotated training data in order to deliver competitive results. As a consequence, significant performance increase can be achieved as soon as faster hardware and higher amount of training data become available [1].

In this chapter, Section 6.2 illustrates the performance evaluation metrics for both audio and text data. Section 6.3 display the comparison of the experimental results of our previous versions with our final versions of the models. Then, interrupting the results, discussing if the objectives were achieved or not, and describing the limitations are placed in Section 6.4. At the end of the chapter, conclusion is given in Section 6.5.

6.2 Performance evaluation metrics

Using different metrics for performance evaluation led to increase the validation of our models and ensure that the models are not memorizing but learning and can predict correctly on new data. In Section 6.2.1 The used performance metrices in audio model are described and in Section 6.2.2 The used performance metrices in text models are described.

6.2.1 Audio model

Due to the small amount of training data which are 1512 for the English datasets, and 954 for the Arabic dataset, that makes us sensitive to overfitting. For audio datasets We used three metrics to evaluate the performance of our models, to end up with the best versions.

Our main focus was on the learning curves as a performance metric to enhance the model, as shown in Figure 6.1. Learning curves are deemed effective tools for monitoring the performance of workers exposed to a new task [2]. It assisted us in determining whether our model was overfitting.

Figure VI.1 Learning curve

There are a lot of things that affect the learning curve result, but we focus on the number of epochs and the learning rate. The way to identify overfitting in the learning curve is when the validation loss decreases until a turning point is reached, at which time it begins to increase, to prevent overfitting at that moment, as in Figure 6.2, we used Earlystopping class from keras.callbacks to stop the training and determine the ideal number of epochs.

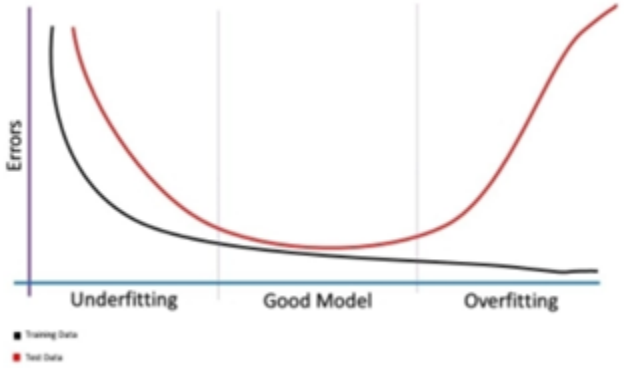

Figure VI.2 Underfitting, good model, and overfitting in Learning curve

In order to reduce the learning rate after the metric had stopped improving, we utilized the ReduceLROnPlateau class from keras.callbacks. We also used the Adam optimizer, a well-known adoptive learning rate optimizer for deep learning.

Furthermore, we also considered the confusion matrix, which is an N x N matrix used for evaluating the performance of a classification by comparing the actual target values with those predicted by the model, as in Figure 6.3.

Figure VI.3 Confusion matrix

One last technique to evaluate the effectiveness of our model is to record our voices for four seconds while we attempt to express a specific emotion to see how the model would classify it.

6.2.2 Text model

The performance evaluation metrics are the key components that identify the model correction and acceptance. A common evaluation metric for classification issues is accuracy. It represents the proportion of all observations to correct predictions. It is an important easy to compute metric that help on choosing the right model. However, we can not depend on the accuracy alone because it need to be with a balanced data, in addition, accuracy is not stable. In our case, the Arabic data and English data are imbalanced So, depending on the accuracy alone might be unreliable on evaluating our model.

To be able to increase the reliability of the model's validation, many evaluation metrics is taking into consideration. The confusion matrix that has been illustrated in 6.2.2 is one of them. In addition to the classification report metrices which include the recall or sensitivity which is the percentage of accurately anticipated positive observations to all of the actual class observations, formula is giving in Figure 6.4 [3]. It gives the true positive rate that its ideally be 1 for a good classifier.

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$= \frac{True\ Positive}{Total\ Actual\ Positive}$$

Figure VI.4 Recall formula

In addition to the precision, which is the proportion of accurately predicted positive observations to all predicted positive observations, formula is giving in Figure 6.5 [3].

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$= \frac{True\ Positive}{Total\ Predicted\ Positive}$$

Figure VI.5 Precision formula

And the F1 score which is the weighted average of precision and recall, both false positives and false negatives are considered while calculating this score. It is not as easy as accuracy to understand, however its more useful when dealing with uneven class distribution. Formula is giving in Figure 6.6 [3].

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

Figure VI.6 F1 score formula

6.3 Experiment results

Reaching the final correct and accurate model is a result of many failures tries. The models that we have tried will be explained in Section 6.3.1 for audio dataset and in Section 6.3.2 for text dataset.

6.3.1 Audio model

The Arabic and English audio datasets both have various issues. One of these is that both have small training data, which could cause overfitting. Moreover, both have an unbalanced audio file of emotions.

In the final model, we used EarlyStopping and ReduceLROnPlateau classes, which helped us determine the optimal number of epochs and reduce the learning rate after the metric had stopped improving. If we compare our final two models with the past models, let's say models A for both English and Arabic datasets (without using Earlystopping), models B for both English and Arabic datasets (without using ReduceLROnPlateau), and models C is the final model (using both EarlyStopping and ReduceLROnPlateau).

In models A, we had to assign a number for epochs, we chose 50, which is recommended by developers. Because of that, the learning phase took a long time to pass the training dataset 50 times. For the confusion matrix in both English and Arabic datasets as shows in Figure 6.9 and Figure 6.10, it gives us a

fine result, but when we see the learning curve in Figure 6.7 and Figure 6.8, the validation has a lot of vibration, that means our model overfit the training dataset by learning even the noise. After all, models A for both languages cannot generalize since it will perform badly on new data.

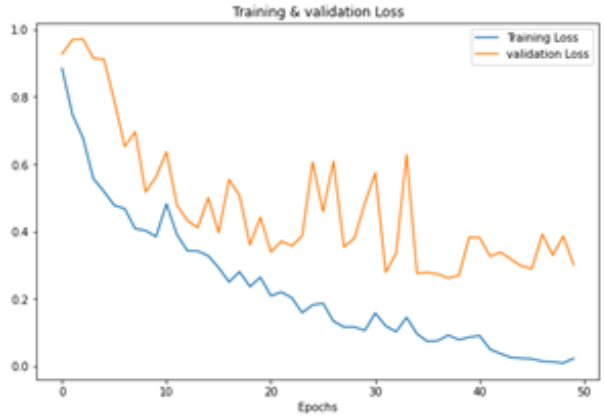

Figure VI.7 Learning curve of model A for English dataset

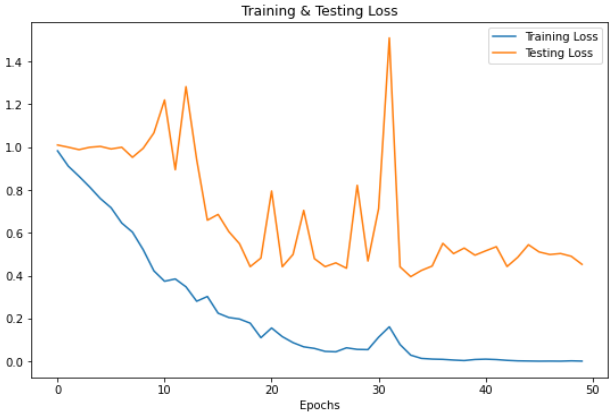

Figure VI.8 Learning curve of model A for Arabic dataset

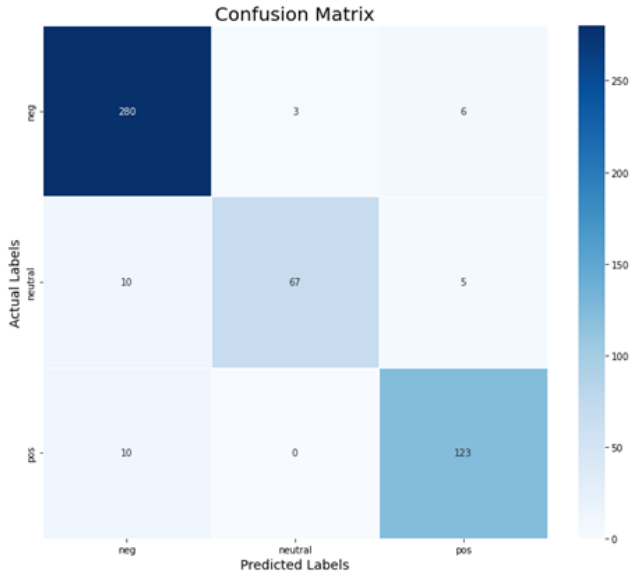

Figure VI.9 Confusion matrix of model A for English dataset

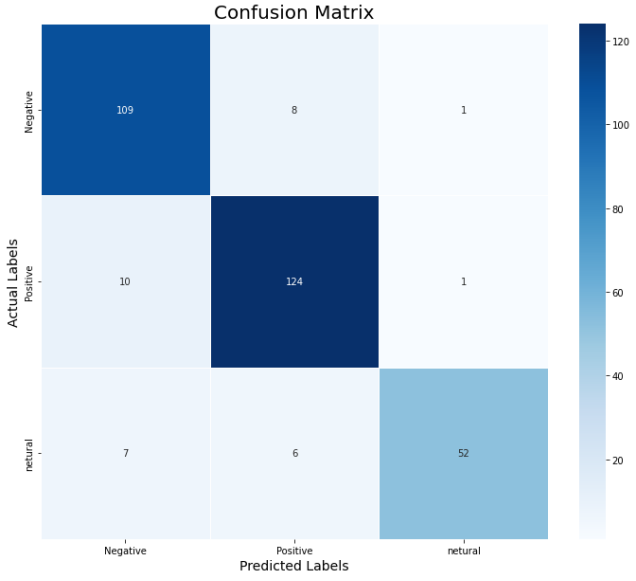

Figure VI.10 Confusion matrix of model A for Arabic dataset

In models B we didn't use ReduceLROnPlateau. For both English and Arabic datasets, we can see in Figure 6.11, Figure 6.12, Figure 6.13, and Figure 6.14, that it is poor by looking at both the learning curve and the confusion matrix, and for the learning curve, we can see that there is a gap between validation and training, which tells us there may be an underfit. Finally, models B cannot capture the underlying trend of the data.

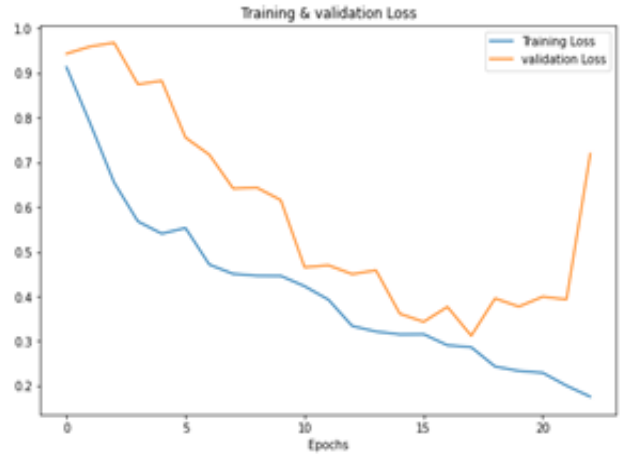

Figure VI.11 Learning curve of model B for English dataset

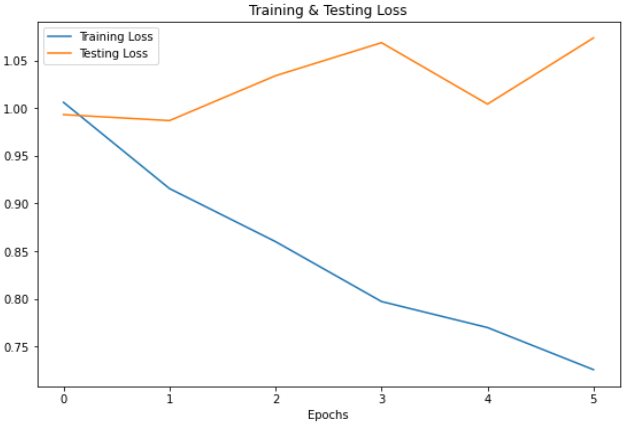

Figure VI.12 Learning curve of model B for Arabic dataset

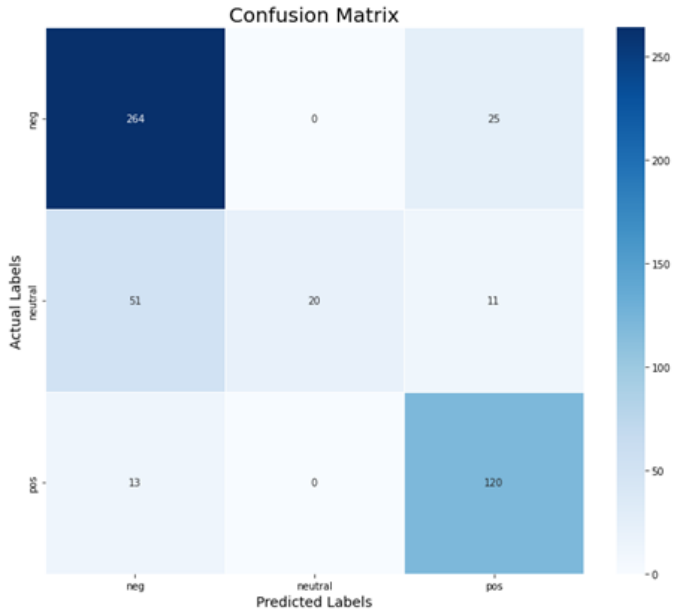

Figure VI.13 Confusion matrix of model B for English dataset

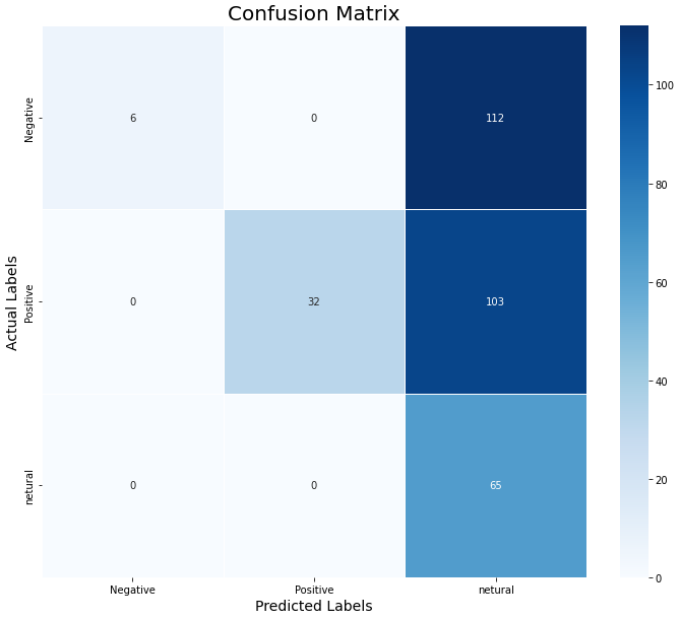

Figure VI.14 Confusion matrix of model B for Arabic dataset

In models C, we used both ReduceLROnPlateau for the English dataset we end up with an accuracy of 85% and works well for positive and negative emotions according to the confusion matrix in Figure 6.17 and learning curve in Figure 6.15. For the live test we tried to express different emotions, in the first recording we tried to yell, "Give me my coffee," in an angry tone. The model correctly classifies this as a

negative in Figure 6.19. In the second recording, we tried to express happiness by asking, "Hey, can I have my coffee?", with a happy and polite tone. The model correctly classifies this as a positive in Figure 6.20. Additionally, we tried our best to express neutral emotion, but the model kept classifying it as negative. Moreover, for the Arabic dataset we end up with an accuracy of 79%, and it work bad classifying emotions based on live test, for the confusion matrix in Figure 6.18 it shows that the model tends to classify the emotions as negative, and for the learning curve it represent an overfit as shows in Figure 6.16.

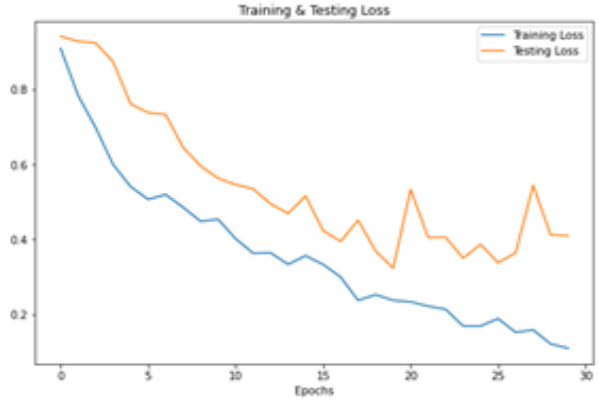

Figure VI.15 Learning curve of model C for English dataset

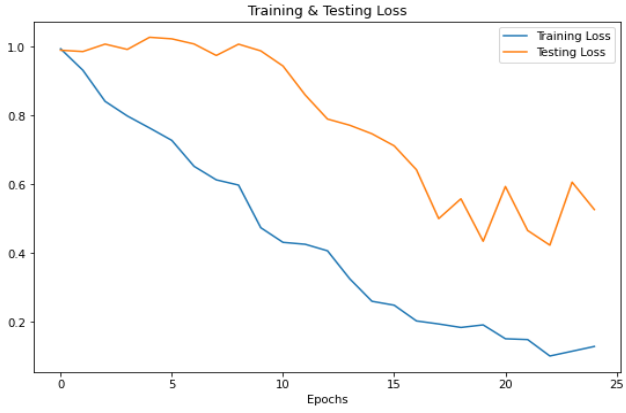

Figure VI.16 Learning curve of model C for Arabic dataset

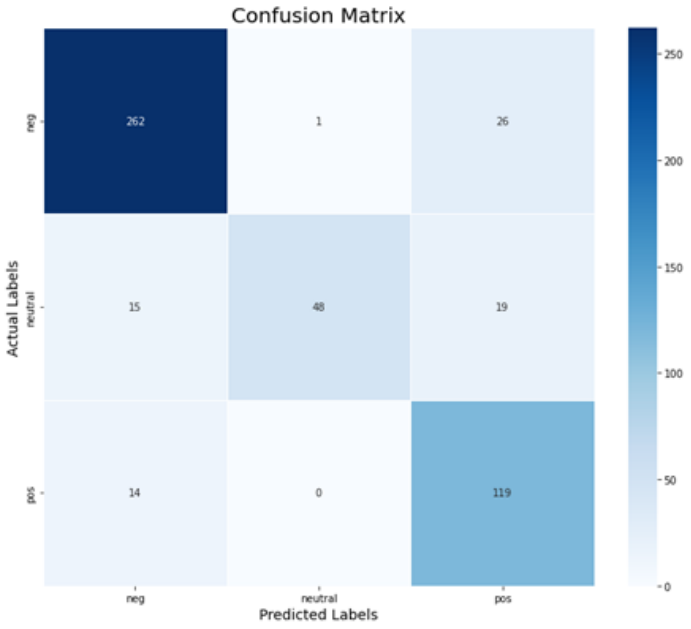

Figure VI.17 Confusion matrix of model C for English dataset

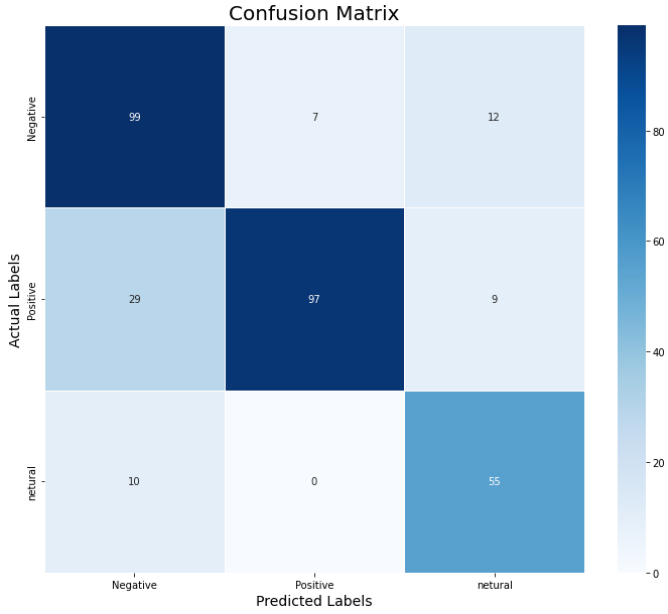

Figure VI.18 Confusion matrix of model C for Arabic dataset

```
1/1 [==============================] - 0s 106ms/step
[['neg']
 ['neg']
 ['neg']]
```

Figure VI.19 First live test of model C for English dataset

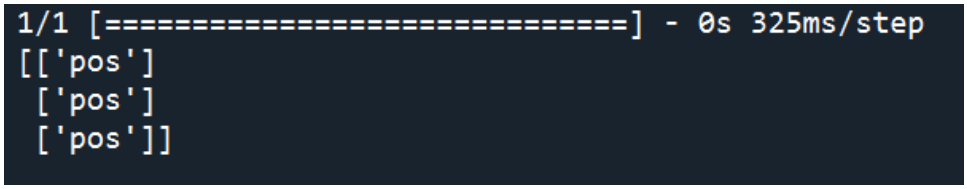

Figure VI.20 Second live test of model C for English dataset

6.3.2 Text model

In our experiment on the text data set we wanted to obtain the best model among each data set Arabic and English by trying different feature extraction techniques and different models on those techniques.

In the beginning we have focus on the "accuracy" as a performance metrics, we were trying to increase the accuracy by determining the effect of using different feature extraction techniques and changing the model between the Arabic data set and the English data set

In the first model Arabic data set was preprocessed by using TF-IDF while English data was prepared using word to sequence Padding and both of the datasets was using **model A**

We got an accuracy of 11.50% for the English dataset and 25.08% for the Arabic dataset we have noticed that the Arabic data set has better accuracy than the English by using the same model and different extraction feature techniques.

Thus, we changed the feature extraction technique for the English data as One hot encoding which gave us a higher accuracy 11.62%, we decided to change the CNN model itself to understand what has affected the dataset accuracy, so we made a new **model B**.

Trying the new model with one hot encoding for the English data gave us an accuracy of 72.00%, we decided to try this new model using the TF-IDF for the Arabic dataset which gave us an accuracy of 75.00%

Trying the new model with TF-IDF gave us higher accuracy than One-hot encoding. So, we used the new model with TF-IDF to make our results more optimal. This gave us an accuracy of 76.00%. the results of the whole models are showed in Table 6.1.

Data set	Model	Feature extraction technique	Accuracy
English data set	A	Word to sequence	11%
Arabic data set	A	TF-IDF	25%
English data set	A	One hot encoding	11%
English data set	B	One hot encoding	72%
Arabic data set	B	TF-IDF	75%
English data set	B	TF-IDF	76%

Table VI.1 accuracy table for text datasets

To make our result more accurate we decided to use the classification report and the confusing matrix on our new model results. We started with the one hot encoding for the English dataset which gave the result shown in Figure 6.22.

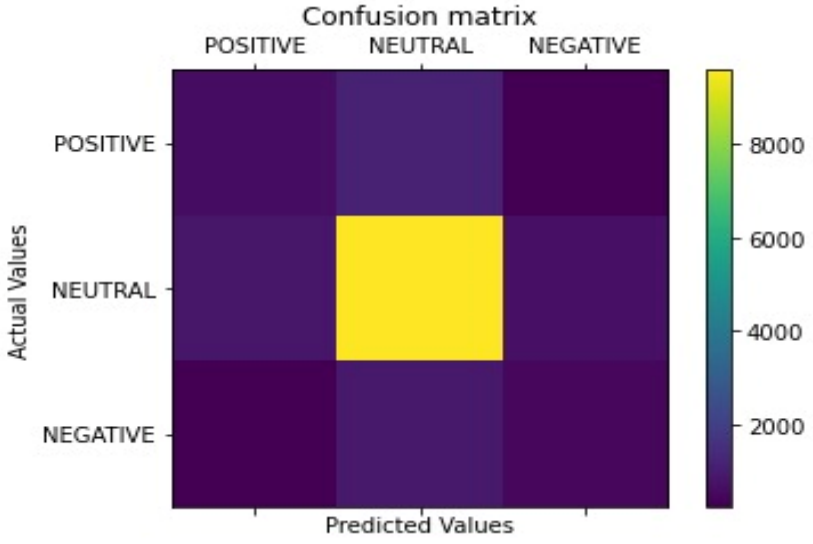

Figure VI.21 confusion matrix for one hot encoding English data set

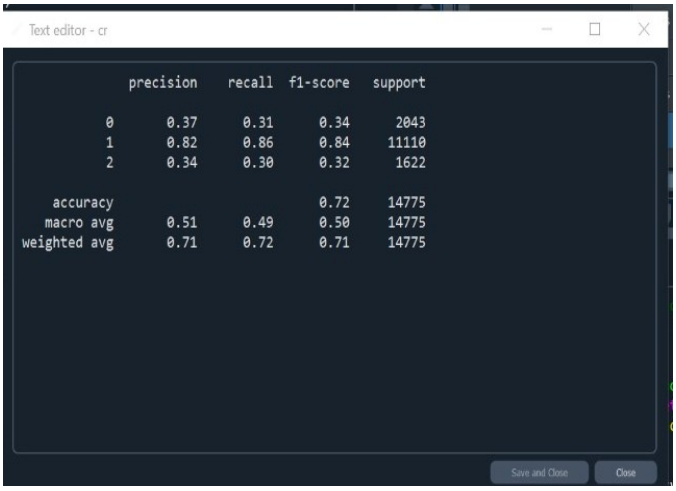

Figure VI.22 classification report for one hot encoding English data set

The confusion matrix results in Figure 6.21 clearly showed that the data was unbalanced, and the neutral label overshadowed the rest of the labels.

After that we calculate the confusion matrix for the English data set with TF-IDF feature extraction technique and it gave the results showing in Figure 6.24.

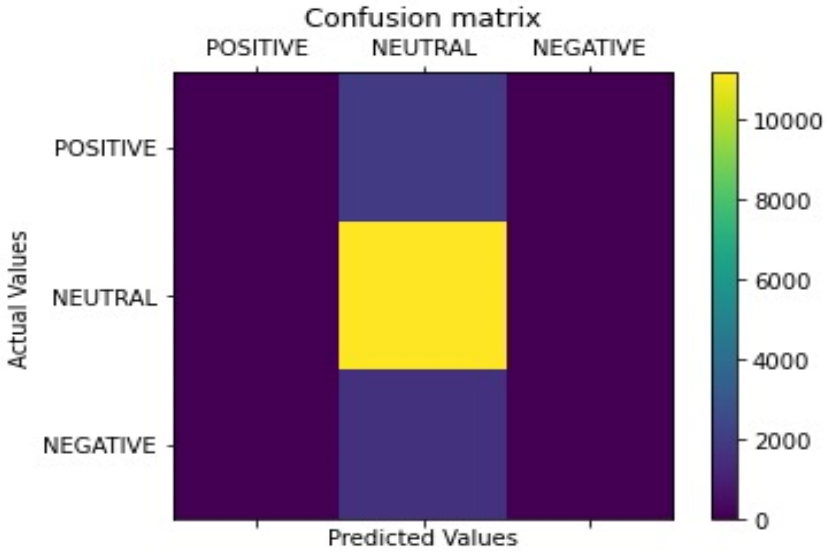

Figure VI.23 confusion matrix for TF-IDF English dataset

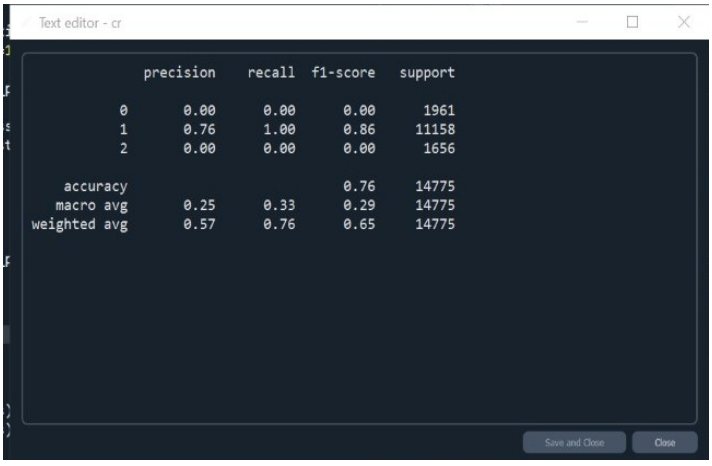

Figure VI.24 classification report for TF-IDF English dataset

Figure 6.24 shows us that the model couldn't predict the rest of the feelings. For the Arabic dataset with TF-IDF we got the results shown in Figure 6.25 which reveal that the model was only able to predict the positive label.

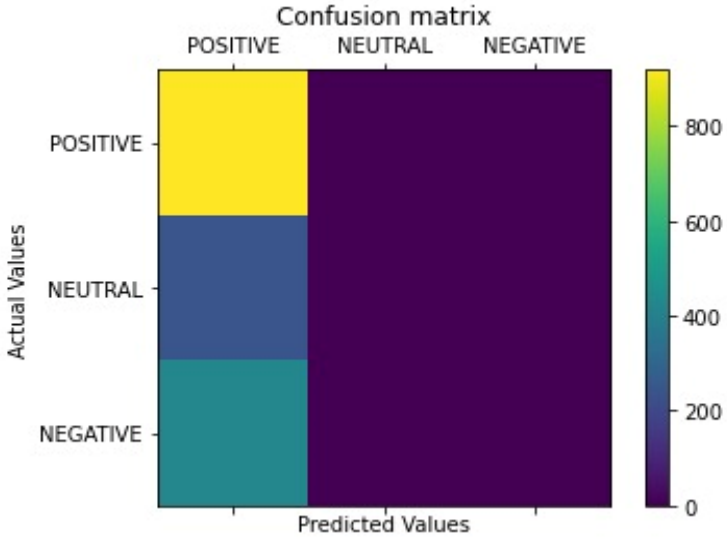

Figure VI.25 confusion matrix for TF-IDF Arabic data set

Those results prove that even if the accuracy of using TF-IDF feature extraction technique was higher than One hot encoding, the model couldn't predict the rest of the feelings unlike when using one hot encoding, and it also shows us that the data was unbalanced in the first place.

6.4 Discussion

In this section the final model for both audio and text datasets with a discussion of either the model reaches the objectives or not, along with the limitation of each model. Section 6.4.1 is talking about audio models and Section 6.4.2 is about the text models.

6.4.1 Audio model

Due to the small amount of training data in both English and Arabic datasets, and the advancement of deep learning, we want the model to generalize rather than overfit. That is why we tried to utilize all available techniques (Earlystopping, ReduceLROnPlateau, dropout layer, etc.) to achieve that objective.

Furthermore, for English dataset the model works well classifying positive and negative emotions, but it performs poorly when it comes to classifying neutral emotions, and this is due to the lack of neutral emotion audio files, as is illustrated in Figure 6.26. After all, our objectives for English dataset are not totally satisfied.

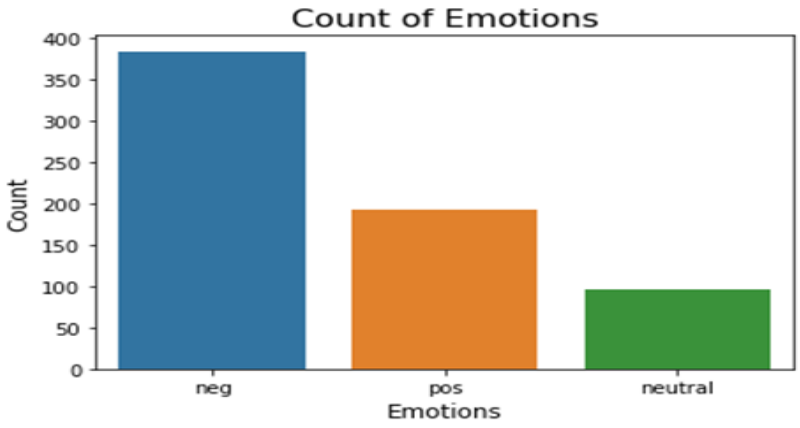

Figure VI.26 Count of emotions for English dataset

Moreover, in general for Arabic dataset we failed to achieve most of our goals for the model, and this happened for several reasons, including that the audio files of emotions are unbalance as shows in Figure 6.27, and the fact that the dataset itself not the best choose.

Also, we tried to do a live test, but the model keep classifying all emotions as negative emotions as we can see in Figure 6.18, that the classification model tends to choose the negative emotions.

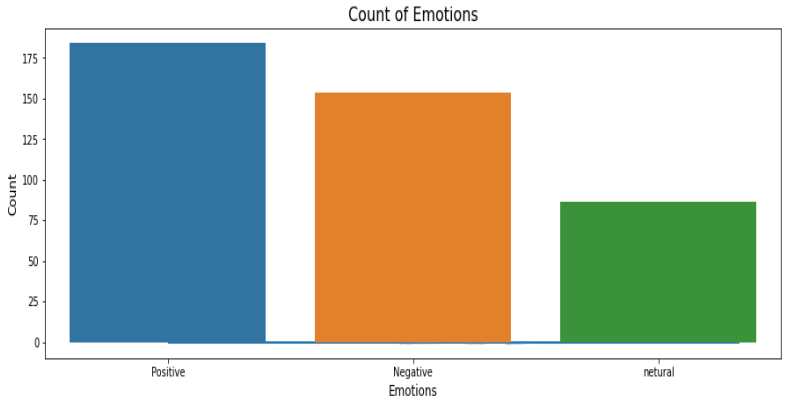

Figure VI.27 Count of emotions for Arabic dataset

At last, both audio datasets share some similar limitations, such as the lack of a large dataset, being only tested on short live audio files and the unbalanced audio file of emotions, but we can say that the model that work with English dataset give us a good performance compare with model of Arabic dataset, which kept classifying all emotions as negative emotions.

6.4.2 Text model

Trying to reach the optimal model is a series of steps and experiments. On the final results, we adopted the **model B** for both Arabic and English data sets however, for the feature extraction on the English data set we rely on two different techniques.

Starting with the Arabic dataset, the used feature extraction method is TF-IDF with 75% accuracy. Even though it's got a high accuracy, the classification report included precision, recall, and F1 score could not be determined which abstract the evaluation of the model performance. On the other hand, a confusion matrix is determined with a high prediction of the positive emotion only. The classification results of this model were not the best, but it achieves a part of our objective of the model by classifying the positive emotion from the rest of data.

One of the biggest limitations when dealing with the Arabic data is the problematic issues with feature extractions techniques. We first apply the word2vector method which requires creating a bag of word. We download two CBOW word to vectors models to get the features of the Arabic data set the first model was created using twitter vocabulary and the second one was made using Wikipedia those models have over 1milion vectors [4], however even when using those models, they could not vectorize all the words in our data set because Arabic language considered as a rich and complex language.

Another limitation we have faced that we could not ably the one hot encoding over the Arabic dataset that effected the comparing of our results between Arabic and English data.

In the English data set we made two different featured extraction methods to compare our results one hot encoding and TF-IDF the last one gave us higher accuracy with 76%, while using one hot encoding gave us 72%. Those results made us though that the higher accuracy means better classification, we used the classification report to make our results more accurate, that gave us a better vision among the data and we understood that the data wasn't balanced in the first place and even that the accuracy of the TF-IDF was higher the model was able to predict the neutral feeling only.

After using the classification report we concluded that the accuracy wasn't suitable in our cases because of the unbalance in the data.

6.5 Conclusion

The main objective of this experiment was to classify feelings in both audio and text data. We didn't quite get the exact desired result, but we did get some interesting results. We went into details about the performance evaluation metrics for the Arabic and English audio and text datasets in section 6.2, compared the experiment findings in section 6.3, and talked about the difficulties we encountered in section 6.4.

Chapter VII : Conclusion and future work

7.1 Introduction

Dealing with emotions is such a complicated process even for the real human being. Simulating the human minds analysis for different emotions is one of the deep learning algorithms' applications. Many researchers were examining different models to try to reach to the optimal model that classify and recognize each emotion correctly. In this project many models were experienced with our datasets until the final model is reached. While experience different models, many obstacles were encountered. Each project has some limitations that cannot be handled at the time due to machine's capacity and performance, lack of data and resources, might be beyond current scientific capabilities and many other barriers.

In this chapter, Section 7.2 defines the final conclusion that we reach in our project. Section 7.3 illustrates the limitations and difficulties we have faced during models' buildings and executions. At the end of the