Analysing emotional content in text has been a challenging task for artificial intelligence considering there are many emotions and datasets typically have imbalanced data across the emotion labels. In addition, humans tend to use ambiguity in textual conversations that can make it even more difficult for AI to interpret the emotional meaning. In the current study, only a few emotions such as happiness, sadness, worry and neutral will be explored to simplify the analysis and to better manage class imbalance. The aim is to investigate if natural language processing can classify the emotional content in text.

**Method**

The dataset was downloaded from Kaggle and consists of social media tweets and the emotional content associated with the tweets. Python 3 using Jupyter notebook text editor was used to analyse the data. The main packages used were pandas, NumPy, Sklearn, tensorflow and Keras. The dataset consisted of 40,000 records (tweets) and 13 emotional labels such as happiness and sadness.

**Data preparation**

The project was split into three separate analyses. The main steps for each analysis:

* Simplify the target variable by reducing the number of classes
* Tokenize the tweets so each word had a corresponding integer.
* Sequence the sentences into lists, so each list consisted of a set of integers which corresponded to different words.
* Post padding so each tweet had the same length.
* Data was split into a training, validation, and test split.
* Neural network was built followed by fitting and compiling the model
* Model was evaluated on the test set

**Model 1 structure**

Model 1 consisted of two classes: happiness and sadness and there were 5209 data points belonging to happiness class and 5165 data points corresponding to the sadness class. The top 15,000 most common words were used for the vocabulary size. The training set and validation set both had 3890 samples while the test set had 2594 samples.

**Model 1 neural network architecture**

Keras library was used to build the neural network and the Sequential API was used to stack the layers. The network consisted of four blocks: the first block’s first layer was an embedding layer consisting of an input size of 15,000 and output size of 25. Batch normalization was followed, and this was then followed by a 1-dimensional global average pooling layer. The second block’s first layer was a fully connected layer consisting of 25 nodes and the Rectified linear activation function (ReLU) was used, and This was followed by batch normalization. The third block’s first layer was a fully connected layer consisting of 12 nodes and ReLU activation function was used. The last block was the output layer consisting of 1 node and the sigmoid activation function was used.

The model was compiled with Adam optimizer with a learning rate of 0.001 and the binary cross entropy function was used. The model was then fitted with the training and validation data, 100 epochs and a batch size of 32. Early stopping was used if validation accuracy didn’t decrease in 30 epochs and learning rate would reduce by a factor of 0.90 if validation loss didn’t decrease in 3 epochs.

**Model 2 structure**

Model 2 consisted of four classes: happiness, sadness, worry and neutral. There were 8638 samples in the neutral class, 8459 samples in the worry class, 5209 samples in the happiness class and 5165 samples in the sadness class. The top 15,000 most common words were used for the vocabulary size. The training set had 15000 samples, the validation set had 5000 samples and the test set had 7471 samples.

**Model 2 neural network architecture**

Keras library was used to build the neural network and the Sequential API was used to stack the layers. The network consisted of five blocks: the first block’s first layer was an embedding layer consisting of an input size of 15,000 and output size of 2000. Batch normalization was followed, and this was then followed by a 1-dimensional global average pooling layer. A flattening layer was then used to transform the data to 1-dimension. The second block’s first layer was a fully connected layer consisting of 30 nodes and the ReLU activation function was used. This was followed by batch normalization and then a dropout layer of 0.5 was used. The third block’s first layer was a fully connected consisting of 10 nodes and the ReLU activation function was used. This was followed by batch normalization and then a dropout layer of 0.3 was used. The fourth block’s first layer was a fully connected layer consisting of 5 nodes and the ReLU activation function was used. This was followed by batch normalization and a dropout rate of 0.1 was used. The last block was a fully connected layer consisting of four nodes and SoftMax activation function was used.

The model was compiled with Adam optimizer with a learning rate of 0.001 and the categorical cross entropy function was used. The model was then fitted with the training and validation data, 100 epochs and a batch size of 32. Early stopping was used if validation accuracy didn’t decrease in 30 epochs and learning rate would reduce by a factor of 0.90 if validation loss didn’t decrease in 3 epochs.

**Model 3 structure**

Model 3 consisted of three classes: happiness, sadness, and love. There were 5209 samples in the happiness class, 5165 samples in the sadness class and 3842 samples in the love class. The top 15,000 most common words were used for the vocabulary size. The training set had 7108 samples, the validation set had 3554 samples and the test set had 3554 samples.

**Model 3 neural network architecture**

Keras library was used to build the neural network and the Sequential API was used to stack the layers.

The network consisted of five blocks: the first block’s first layer was an embedding layer consisting of an input size of 15,000 and output size of 2000. Batch normalization was followed, and this was then followed by a 1-dimensional global average pooling layer. A flattening layer was then used to transform the data to 1-dimension. The second block’s first layer was a fully connected layer consisting of 30 nodes and the ReLU activation function was used. This was followed by batch normalization and then a dropout layer of 0.5 was used. The third block’s first layer was a fully connected layer consisting of 10 nodes and the ReLU activation function was used. This was followed by batch normalization and then a dropout layer of 0.3 was used. The fourth block’s first layer was a fully connected layer consisting of 5 nodes and the ReLU activation function was used. This was followed by batch normalization and a dropout rate of 0.1 was used. The last block was a fully connected layer consisting of three nodes and SoftMax activation function was used.

The model was compiled with Adam optimizer with a learning rate of 0.001 and the categorical cross entropy function was used. The model was then fitted with the training and validation data, 100 epochs and a batch size of 32. Early stopping was used if validation accuracy didn’t decrease in 30 epochs and learning rate would reduce by a factor of 0.90 if validation loss didn’t decrease in 3 epochs.

**Results**

*Table 1.* Training, validation, and test accuracy for each model.

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Training | Validation | Test |
| Model 1 | 99% | 77% | 78% |
| Model 2 | 92% | 38% | 37% |
| Model 3 | 96% | 57% | 57% |

As can be seen from table 1, model 1 achieved the highest training, validation and most importantly, test accuracy. Model 2 clearly overfitted the training data and could not generalize to the validation and test sets. Model 3 did better than model 2 but there was still a major overfitting problem.

**Discussion**

The aim of the current study was investigating whether natural language processing could classify the emotional content of tweets and this was tested using a neural network. Model 1 performed well considering it had an accuracy of 78% on classifying the emotional context of tweets as happiness or sadness in the test set. Model 2 extended on model 1 and added worry and neutral classes but performed poor at classifying the emotional context of tweets into their correct classes on the test set and clearly overfitted the training data. Model 3 included happiness, sadness and love classes and performed better than model 2 but still clearly overfitted considering it scored 57% on the test set. One limitation of the current research was model 2 had a slight class imbalance issue with the happiness and sadness classes having noticeably less samples than neutral and worry classes which may have been one of the reasons for the poor classification accuracy. Considering only a subset of classes were selected in this study to explore, future research could extend the multi-class classification and include more emotions labels. Future research should also attempt to have near equal number of samples for each class.

In conclusion, the neural network used in the research performed well at classifying the emotional content of tweets into happiness or sadness classes. Adding additional classes to model 2 and model 3 resulted in poor performance and major overfitting was present.