REPORT ON EMOTION INTENSITY

1 - CALCULATION OF EMO-INT WITHOUT USING DEEP LEARNING

1. Data Overview:

The dataset used in this analysis, contains textual data accompanied by emotion labels ranging from 0 to 5. It offers a diverse range of expressions reflecting different emotions.

2. Data Preprocessing:

After loading the dataset into a pandas DataFrame, sentiment analysis was conducted using the VADER sentiment analyzer. The resulting sentiment scores were then scaled between 0 and 1 to standardize the intensity across all entries. Furthermore, emotion labels were mapped to their respective emotion names for better interpretability.

3. Emotion Distribution:

To visualize the distribution of emotions within the dataset, a bar plot was created. Each emotion label was depicted on the x-axis, while the corresponding count of texts expressing that emotion was represented on the y-axis. This visualization provided insight into the prevalence of various emotions in the dataset.

4. Emotion Intensity Analysis:

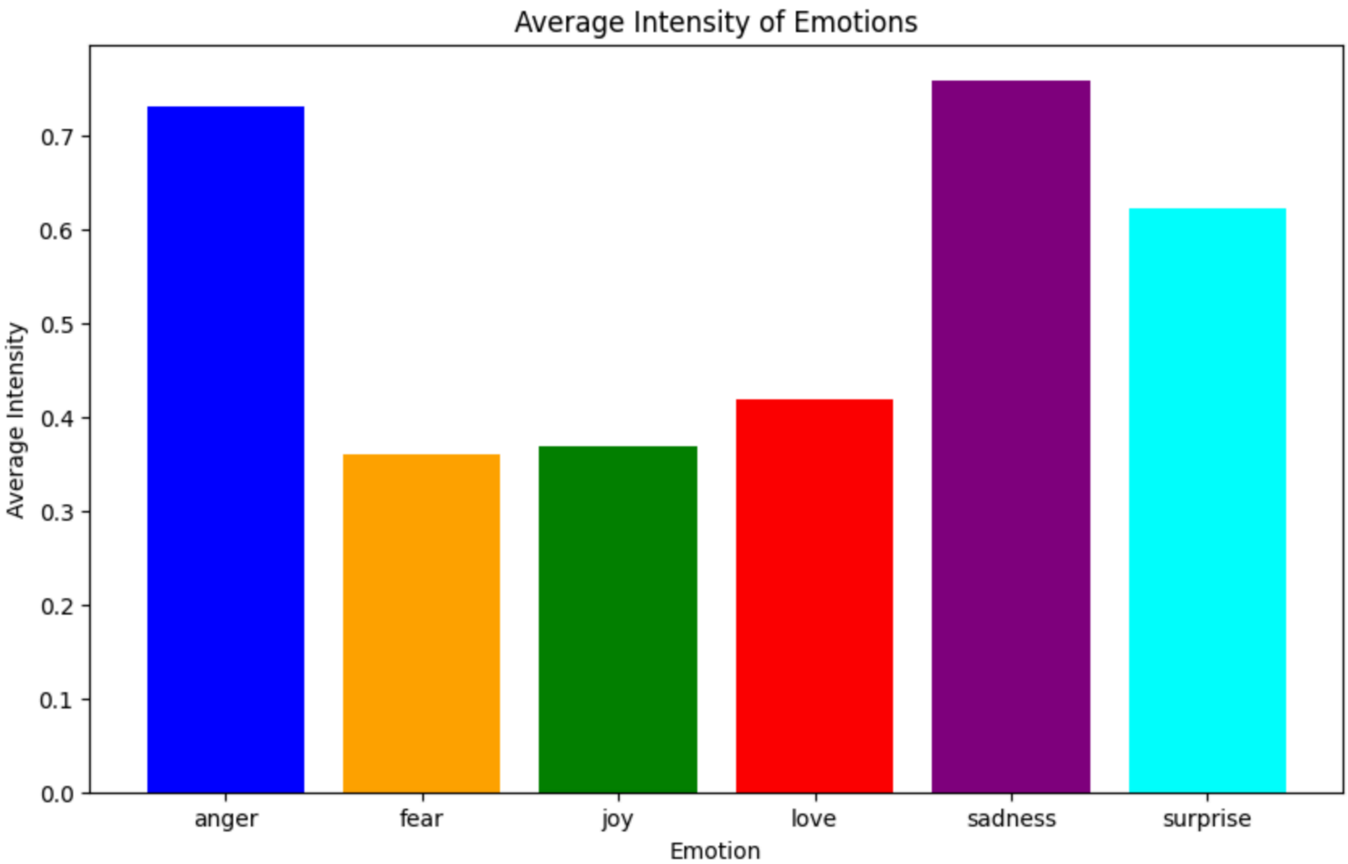
A deeper understanding of the emotional content was gained by calculating the average intensity of each emotion based on the scaled sentiment scores. A grouped bar plot was generated to illustrate the average intensity of emotions, highlighting variations in emotional expression across different categories.

5. Conclusion and Insights:

The analysis uncovered a diverse range of emotional expressions captured within the dataset, reflecting the intricate nature of human emotions. While certain emotions appeared more frequently than others, the intensity analysis revealed nuanced variations in the strength of emotional sentiment across different categories. These insights offer valuable information for understanding the emotional landscape of the dataset and may inform various applications, including sentiment analysis and emotion recognition.

```
                                                    text  emotion  \
0                                i didnt feel humiliated      joy
1      i can go from feeling so hopeless to so damned...      joy
2       im grabbing a minute to post i feel greedy wrong     fear
3      i am ever feeling nostalgic about the fireplac...    anger
4                                   i am feeling grouchy     fear
...                                                  ...      ...
15995  i just had a very brief time in the beanbag an...      joy
15996  i am now turning and i feel pathetic that i am...      joy
15997                     i feel strong and good overall  sadness
15998  i feel like this was such a rude comment and i...     fear
15999  i know a lot but i feel so stupid because i ca...      joy

       emotion_intensity
0               0.631076
1               0.541525
2               0.164728
3               0.564840
4               0.327119
...                  ...
15995           0.460609
15996           0.209326
15997           0.873216
15998           0.683294
15999           0.096764
```



2 - WITH USING DEEP LEARNING ALGORITHMS

Implementation of a text classification model using deep learning techniques to recognize emotions in textual data. Emotion recognition plays a crucial role in various applications such as sentiment analysis, customer feedback analysis, and chatbots. This report provides a comprehensive overview of the code, detailing the deep learning techniques used and their implementation.

Data Preprocessing:

- **Data Loading:** The dataset is loaded from the file 'training.csv' using the pandas library. The dataset consists of text samples and corresponding emotion labels.
- **Text Tokenization:** The text data is tokenized using the Tokenizer class from the TensorFlow Keras library. Tokenization is the process of converting text into numerical tokens. The tokenizer is fitted on the text data to generate a vocabulary of words.
- **Padding Sequences:** To ensure uniform input dimensions for the neural network, the tokenized sequences are padded to a maximum length of 100 tokens using the pad_sequences function. Padding is necessary because neural networks require inputs of fixed dimensions.

Model Architecture:

- **Embedding Layer:** The tokenized sequences are passed through an Embedding layer. This layer converts each token into a dense vector representation. The output_dim parameter specifies the dimensionality of the embedding space, and the input_length parameter specifies the length of input sequences.
- **LSTM Layer:** Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture. It is well-suited for processing sequential data like text because it can capture long-range dependencies and handle vanishing gradient problems.
- **Dropout Layers:** Dropout regularization is applied to prevent overfitting. Dropout randomly sets a fraction of input units to zero during training, which helps to reduce the model's reliance on specific features.
- **Dense Layers:** Fully connected dense layers are used for feature extraction and classification. The number of neurons in the dense layers and activation functions are specified based on the task requirements.

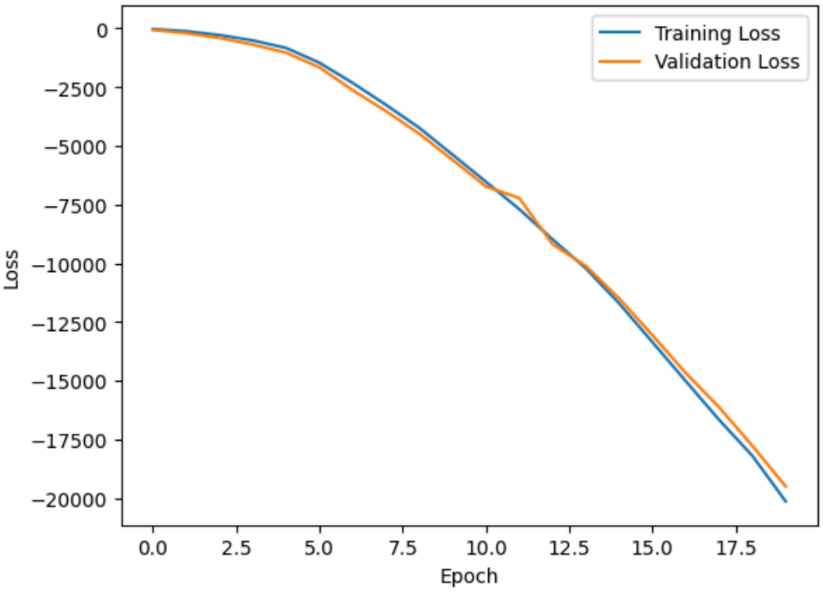
Model Compilation and Training:

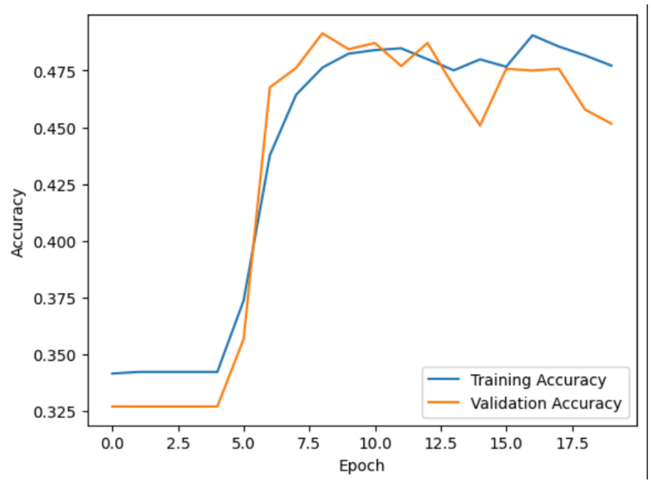
- **Model Compilation:** The model is compiled using the Adam optimizer and binary cross-entropy loss function. Adam is an efficient optimization algorithm for training neural networks. Binary cross-entropy loss is suitable for binary classification tasks.
- **Early Stopping:** Early stopping is employed to prevent overfitting. It monitors the validation loss during training and stops training if the loss does not improve for a specified number of epochs (patience).

- **Training:** The model is trained on the training data with a validation split of 20%. Training involves iteratively updating the model's parameters to minimize the loss function.

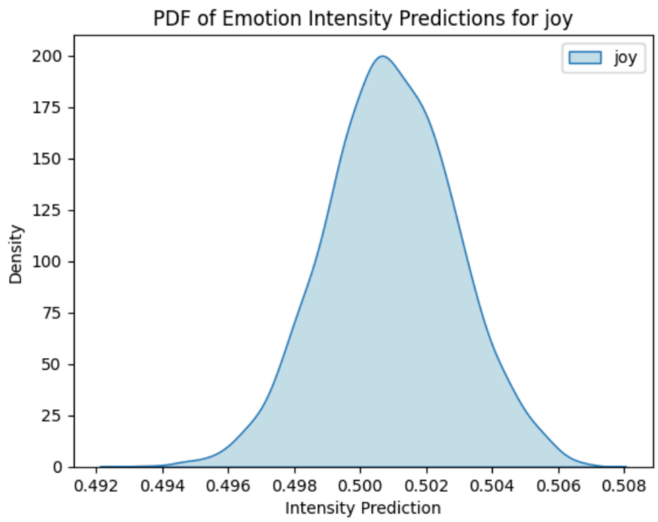
Evaluation:
After training, the model's performance is evaluated on the test data. The evaluate method is used to compute the test loss and accuracy. The test loss quantifies the difference between the true labels and the model's predictions, while the accuracy indicates the proportion of correctly classified samples.
Conclusion:
In conclusion, a text classification model for emotion recognition using deep learning techniques. By preprocessing the data, defining an appropriate model architecture, and training the model on labeled data, it aims to accurately classify text samples into different emotion categories. Further experimentation and optimization may be required to improve the model's performance and generalization capabilities on unseen data.

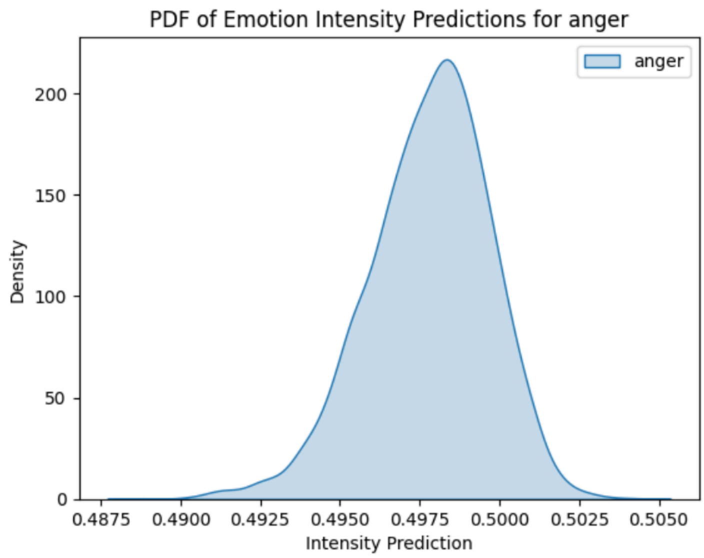
```
Epoch 1/20
160/160 [==============================] - 24s 135ms/step - loss: -26.5436 - accuracy: 0.3415 - val_loss: -67.6954 - val_accuracy: 0.3270
Epoch 2/20
160/160 [==============================] - 19s 120ms/step - loss: -119.6084 - accuracy: 0.3422 - val_loss: -201.1522 - val_accuracy: 0.3270
Epoch 3/20
160/160 [==============================] - 20s 125ms/step - loss: -281.5910 - accuracy: 0.3422 - val_loss: -407.8900 - val_accuracy: 0.3270
Epoch 4/20
160/160 [==============================] - 20s 128ms/step - loss: -511.8813 - accuracy: 0.3422 - val_loss: -687.0214 - val_accuracy: 0.3270
Epoch 5/20
160/160 [==============================] - 19s 119ms/step - loss: -830.1268 - accuracy: 0.3422 - val_loss: -1032.3918 - val_accuracy: 0.3270
Epoch 6/20
160/160 [==============================] - 20s 126ms/step - loss: -1461.0792 - accuracy: 0.3736 - val_loss: -1656.5485 - val_accuracy: 0.3566
Epoch 7/20
160/160 [==============================] - 19s 119ms/step - loss: -2321.3279 - accuracy: 0.4377 - val_loss: -2630.1829 - val_accuracy: 0.4676
Epoch 8/20
160/160 [==============================] - 21s 134ms/step - loss: -3252.9119 - accuracy: 0.4644 - val_loss: -3525.7395 - val_accuracy: 0.4762
Epoch 9/20
160/160 [==============================] - 20s 126ms/step - loss: -4233.3203 - accuracy: 0.4763 - val_loss: -4487.1631 - val_accuracy: 0.4914
Epoch 10/20
160/160 [==============================] - 19s 120ms/step - loss: -5378.0537 - accuracy: 0.4824 - val_loss: -5600.6807 - val_accuracy: 0.4844
Epoch 11/20
160/160 [==============================] - 21s 133ms/step - loss: -6528.0693 - accuracy: 0.4840 - val_loss: -6725.8921 - val_accuracy: 0.4871
Epoch 12/20
160/160 [==============================] - 20s 128ms/step - loss: -7694.2256 - accuracy: 0.4848 - val_loss: -7220.2920 - val_accuracy: 0.4770
Epoch 13/20
160/160 [==============================] - 19s 119ms/step - loss: -8980.2080 - accuracy: 0.4800 - val_loss: -9186.6445 - val_accuracy: 0.4871
Epoch 14/20
160/160 [==============================] - 21s 130ms/step - loss: -10210.6934 - accuracy: 0.4751 - val_loss: -10112.2803 - val_accuracy: 0.4680
Epoch 15/20
160/160 [==============================] - 19s 121ms/step - loss: -11708.0986 - accuracy: 0.4799 - val_loss: -11497.3623 - val_accuracy: 0.4508
Epoch 16/20
160/160 [==============================] - 20s 124ms/step - loss: -13360.4395 - accuracy: 0.4767 - val_loss: -13063.9551 - val_accuracy: 0.4758
Epoch 17/20
160/160 [==============================] - 21s 133ms/step - loss: -15017.1611 - accuracy: 0.4905 - val_loss: -14673.7207 - val_accuracy: 0.4750
Epoch 18/20
160/160 [==============================] - 19s 121ms/step - loss: -16665.0625 - accuracy: 0.4855 - val_loss: -16129.1309 - val_accuracy: 0.4758
Epoch 19/20
160/160 [==============================] - 20s 127ms/step - loss: -18196.7969 - accuracy: 0.4816 - val_loss: -17772.3301 - val_accuracy: 0.4578
Epoch 20/20
160/160 [==============================] - 21s 130ms/step - loss: -20136.7148 - accuracy: 0.4771 - val_loss: -19501.7402 - val_accuracy: 0.4516
100/100 [==============================] - 2s 18ms/step - loss: -19423.1191 - accuracy: 0.4384
Test Loss: -19423.119140625
Test Accuracy: 0.43843749165534973
```

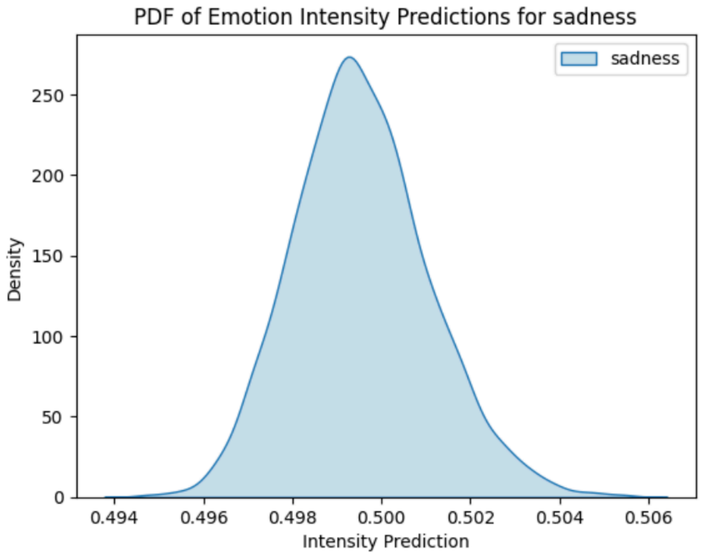


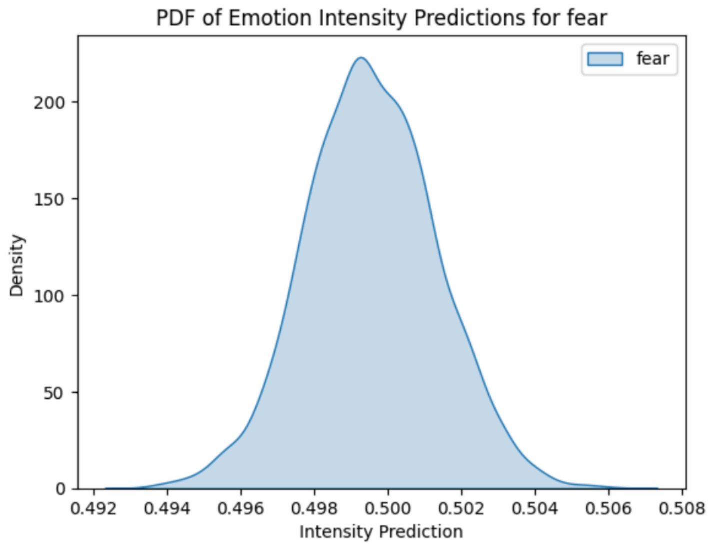


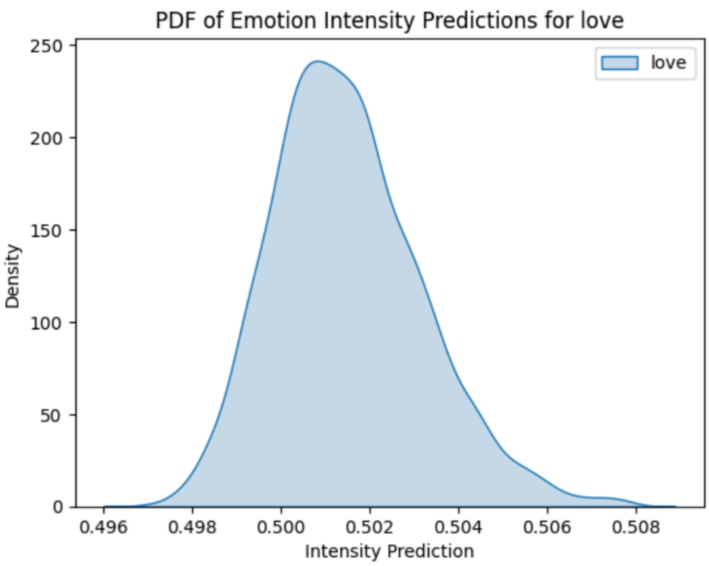

```
Sample 1 – Emotion: joy, Intensity Prediction: 0.5043044686317444
Sample 2 – Emotion: sadness, Intensity Prediction: 0.4984952509403229
Sample 3 – Emotion: joy, Intensity Prediction: 0.49894630908966064
Sample 4 – Emotion: sadness, Intensity Prediction: 0.4983007311820984
Sample 5 – Emotion: joy, Intensity Prediction: 0.5013655424118042
Sample 6 – Emotion: joy, Intensity Prediction: 0.49837997555732727
Sample 7 – Emotion: love, Intensity Prediction: 0.5016492605209351
Sample 8 – Emotion: joy, Intensity Prediction: 0.5016940832138062
Sample 9 – Emotion: anger, Intensity Prediction: 0.49945339560508728
Sample 10 – Emotion: love, Intensity Prediction: 0.4997105300426483
Sample 11 – Emotion: joy, Intensity Prediction: 0.5007681250572205
Sample 12 – Emotion: fear, Intensity Prediction: 0.5019517540931702
Sample 13 – Emotion: sadness, Intensity Prediction: 0.4981066584587097
Sample 14 – Emotion: joy, Intensity Prediction: 0.5051401257514954
Sample 15 – Emotion: joy, Intensity Prediction: 0.5018670558929443
Sample 16 – Emotion: surprise, Intensity Prediction: 0.4977682828903198
Sample 17 – Emotion: fear, Intensity Prediction: 0.5041618943214417
Sample 18 – Emotion: sadness, Intensity Prediction: 0.4992425739765167
Sample 19 – Emotion: anger, Intensity Prediction: 0.49662065505981445
Sample 20 – Emotion: surprise, Intensity Prediction: 0.4989396035671234
```

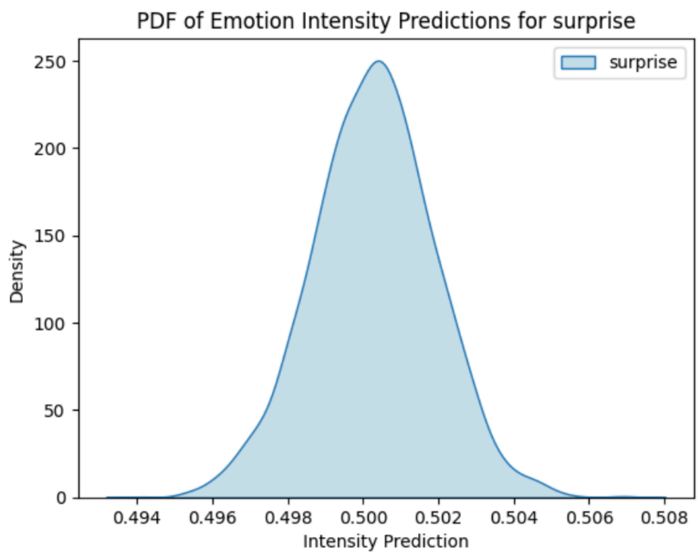

PDF of Emotion Intensity Predictions for joy
joy
Density
Intensity Prediction
0 25 50 75 100 125 150 175 200
0.492 0.494 0.496 0.498 0.500 0.502 0.504 0.506 0.508


PDF of Emotion Intensity Predictions for anger
anger
Density
Intensity Prediction
0 50 100 150 200
0.4875 0.4900 0.4925 0.4950 0.4975 0.5000 0.5025 0.5050


PDF of Emotion Intensity Predictions for sadness
sadness
Density
Intensity Prediction
0 50 100 150 200 250
0.494 0.496 0.498 0.500 0.502 0.504 0.506


PDF of Emotion Intensity Predictions for fear
fear
Density
Intensity Prediction
0 50 100 150 200
0.492 0.494 0.496 0.498 0.500 0.502 0.504 0.506 0.508


PDF of Emotion Intensity Predictions for love
love
Density
Intensity Prediction
0 50 100 150 200 250
0.496 0.498 0.500 0.502 0.504 0.506 0.508


PDF of Emotion Intensity Predictions for surprise
surprise
Density
Intensity Prediction
0 50 100 150 200 250
0.494 0.496 0.498 0.500 0.502 0.504 0.506 0.508