**WEEK-2 TASK**

**DOC BY AMEEN KHAN**

**1. Why did you choose this particular dataset for the task? What specific features or attributes made it suitable for emotion detection?**

This dataset was chosen because it contains labeled text samples spanning multiple emotion classes, such as happiness, sadness, anger, and more. Its diversity and size (with 40,000+ records) make it ideal for training a robust emotion detection model. Additionally, its simple structure with text and emotion labels made it straightforward to preprocess and use in deep learning tasks.

**2. How did you ensure that the dataset was representative of all emotion classes? If there were any imbalances, how did you handle them?**

There were all the six emotions as well as more 3 emotions, so the other 3 were removed appropriately. Class distributions were analyzed during EDA to ensure representation across all emotions. Significant imbalances were addressed through upsampling underrepresented classes to match the largest class size. This ensured the model did not bias predictions toward overrepresented classes, enabling fairer performance across all emotion categories.

**3. Did you encounter missing, irrelevant, or noisy data in the dataset? How did you deal with these issues?**

The dataset was checked for missing values, irrelevant text, and noise such as special characters, emojis or slang. Missing data was removed, and text preprocessing techniques like tokenization, lowercasing, and removal of unnecessary characters were applied. Slang and abbreviations were normalized, ensuring clean and meaningful input for the model.

**4. What key patterns or insights did you observe during the exploratory data analysis (EDA)? How did these inform your approach?**

EDA revealed variations in text lengths across emotions and imbalanced class distributions. These findings informed key decisions:

1. Padding and truncating text sequences to ensure consistent input length.
2. Balancing classes through upsampling to improve model performance on underrepresented emotions. These insights helped tailor the preprocessing pipeline and guided training strategies for better results.

### ****5. What preprocessing steps did you take to prepare the text data for training? Why were these steps necessary for this task?****

The preprocessing steps included lowercasing, tokenization, removing special characters, normalizing slang, and handling abbreviations. These steps ensured uniformity, reduced noise, and enhanced the model’s ability to extract meaningful patterns from the text. Consistent formatting is critical for achieving better generalization in machine learning models.

### ****6. How did you handle special cases such as emojis, abbreviations, or slang in the text data? Why did you choose that approach?****

Emojis were replaced with their textual meanings (e.g., 😊 → "happy"), while abbreviations and slang were expanded to their full forms (e.g., "u" → "you"). This approach ensured that these elements were meaningful within the context of emotion detection, preserving their contribution to the classification task.

### ****7. Did you perform any specific transformations on the labels (emotion categories) before training the model? Why?****

Labels were one-hot encoded to represent each emotion category as a binary vector. This transformation was necessary for multi-class classification tasks, allowing the model to output probabilities for each class and enabling better optimization during training.

Label Mapping:

0: anger, 1: fear, 2: joy, 3: neutral, 4: sadness, 5: surprise

### ****8. If your preprocessing pipeline included removing elements (e.g., stopwords or punctuation), how did you decide what to remove and what to keep?****

Stopwords were retained as they could carry important contextual clues for emotions (e.g., "not happy"). Punctuation was mostly removed, except for those essential for context (e.g., "!" for excitement). Decisions were guided by the need to preserve sentiment-relevant information.

### ****9. How did you convert the text into a format that the model could process? Why did you choose that method?****

The text was converted into numerical sequences using a tokenizer that mapped each word to a unique integer. Sequences were then padded or truncated to a fixed length of 150. This method was chosen because it efficiently represents text data for LSTM models while preserving word order and context.

### ****10. Did you face challenges in representing text data numerically? If so, how did you overcome them?****

Yes, challenges included handling out-of-vocabulary words and ensuring consistent sequence lengths. These were addressed by limiting the vocabulary size to 5000 and applying padding/truncation. Pre-trained word embeddings like GloVe further enriched word representations, improving the model's understanding.

### ****11. What trade-offs did you consider when selecting the features or representations for the task?****

The main trade-off was between simplicity and expressiveness. Count-based methods were simpler but lacked context. Pre-trained embeddings like GloVe were more expressive but increased computational requirements. Ultimately, GloVe was chosen to capture semantic relationships, balancing performance and efficiency.

### ****12. How did you determine the split between training and testing data? Why was this split ratio appropriate for this task?****

The dataset was split into 80% training, 10% validation, and 10% testing. This ratio ensured sufficient data for training while reserving enough for unbiased evaluation. The validation set helped monitor overfitting during training and tune hyperparameters effectively.

### ****13. What steps did you take to evaluate and compare different models or approaches for the task? How did you make the final choice?****

Several models were tested, including LSTMs and BiLSTMs with and without pre-trained embeddings. Models were compared based on validation accuracy, loss, and generalization to the test set. The BiLSTM with pre-trained GloVe embeddings was chosen for its balance between performance and efficiency.

### ****14. What challenges did you encounter during training, such as performance issues or overfitting? How did you address them?****

Challenges included slow training and overfitting on the training data. Overfitting was mitigated with dropout layers and early stopping. Performance issues were addressed by using batch normalization and training on a GPU, improving both stability and speed.

### ****15. How did you decide on the hyperparameters or settings for training? Did you experiment with different configurations?****

Hyperparameters like learning rate, batch size, and dropout rate were tuned through experimentation. A learning rate scheduler was employed to adjust the rate dynamically, ensuring smoother convergence. Batch sizes of 64 and dropout rates of 0.3–0.5 were finalized based on validation performance.

### ****16. What evaluation metrics did you use to measure the performance of your approach? Why were these metrics suitable for this task?****

Accuracy, precision, recall, and F1-score were used to measure performance. F1-score was particularly important for addressing class imbalance, as it balances precision and recall, ensuring fair evaluation across all emotion categories.

### ****17. How did you interpret and act on cases where the model misclassified the emotion of a text? What steps did you take to reduce such errors?****

Misclassified cases were analyzed using confusion matrices to identify confusion between similar emotions (e.g., fear and anger). Strategies included enriching the dataset with more diverse samples and fine-tuning the model with class weights to handle imbalances.

### ****18. How did you ensure your model generalized well to unseen data? What indicators helped you assess this?****

Generalization was ensured through regularization techniques (dropout, batch normalization) and monitoring validation loss. Consistency between validation and test metrics was a key indicator of good generalization. Early stopping prevented overfitting to the training data.

### ****19. Did you notice any specific patterns in the errors made by your model? How did these insights inform your next steps?****

The model occasionally confused neutral emotions with sadness or joy due to subtle differences in text tone. This highlighted the need for better context understanding. Next steps included exploring transformer-based models to capture contextual nuances more effectively.