TASK 1 (CLASSIFICATION)

0.1 Problem Statement

0.1.1 Dataset File: Topical_chat .csv

This dataset consists of over 8000 conversations and over 184000 messages within each message, there is a conversation id, which is basically which conversation the message takes place in. Each message is either the start of a conversation or a reply from the previous message. There is also a sentiment, which represents the emotion that the person who sent the message is feeling. There are 8 sentiments: Angry, Curious to Dive Deeper, Disguised, Fearful, Happy, Sad, and Surprised. Sentiment Analysis: Build a multi-class sentiment analysis model based on this dataset. Please report metrics for the model.

SOLUTION

1 DATASET

The dataset(Topical_chat .csv) exhibits class imbalance with eight categories: Angry, Curious to Dive Deeper, Disguised, Fearful, Happy, Sad, and Surprised.

```
<<<<Percentage Distribution of Sentiments>>>>
Curious to dive deeper    42.939806
Neutral                   21.960154
Surprised                 16.264008
Happy                     15.721468
Sad                        1.344673
Disgusted                  0.760194
Fearful                    0.544664
Angry                      0.465035
Name: sentiment, dtype: float64
```

Figure 1. Percentage Distribution of Sentiments

Observing Figure 1, it becomes apparent that the classes Sad, Disguised, Fearful, and Angry have a notably low occurrence.The column ”message” has five records that are missing.

2 DATA PREPORCESSING AND DATA CLEANING

The subsequent actions are taken in this step:

- Eliminating the missing records, given that there are only five instances of such records.
- Converting the text to lowercase
- Removing special characters and digits
- Removing stop words and applying lemmatization

3 APPROACH

Initially, we adhere to the traditional Machine learning algorithmic approach, and subsequently, we employ deep learning techniques.

3.1 Machine Learning Approach

In this approach , we used two algorithms:

- Naive Bayes Classifier
- Rnadomforest Classifier

3.1.1 Naive Bayes Classifier

As Naive Bayes handles both categorical target variable as well as Numerical target variable.I have followed both techniques:In first technique, I have not converted target variable into numerical variable.In the second technique, I have used label encoding to convert target variable into numerical variable.

Result & Metrics: The performance of the algorithm did not improve in both techniques.

Classification Report:

	precision	recall	f1-score	support
Angry	0.00	0.00	0.00	154
Curious to dive deeper	0.45	0.93	0.61	16097
Disgusted	0.00	0.00	0.00	302
Fearful	0.00	0.00	0.00	207
Happy	0.47	0.09	0.15	5942
Neutral	0.44	0.12	0.19	8294
Sad	0.33	0.01	0.01	522
Surprised	0.42	0.07	0.12	6157
accuracy			0.45	37675
macro avg	0.26	0.15	0.14	37675
weighted avg	0.44	0.45	0.35	37675

Accuracy: 0.4509621765096218

Figure 2. Classification Report by using -Technique 1

Classification Report:

	precision	recall	f1-score	support
Angry	0.00	0.00	0.00	154
Curious to dive deeper	0.45	0.93	0.61	16097
Disgusted	0.00	0.00	0.00	302
Fearful	0.00	0.00	0.00	207
Happy	0.47	0.09	0.15	5942
Neutral	0.44	0.12	0.19	8294
Sad	0.33	0.01	0.01	522
Surprised	0.42	0.07	0.12	6157
accuracy			0.45	37675
macro avg	0.26	0.15	0.14	37675
weighted avg	0.44	0.45	0.35	37675

Accuracy: 0.4509621765096218

Figure 3. Classification Report by using -Technique 2

Metrics Explanation :
Term Explanation

- **Precision**: The proportion of correctly identified instances among those labeled as positive. It measures how often the model is correct when it predicts a positive result.
- **Recall**: The proportion of actual positive instances that were correctly identified. It measures how often the model correctly identifies all positive cases.
- **F1-score:** The harmonic mean of precision and recall, providing a balanced measure of accuracy that considers both. Higher F1-scores indicate better overall performance.
- **Support**: The number of instances for each class in the dataset. It shows how much data was available for training and evaluation.

Class-Specific Metrics Explanation:

- **Angry, Disgusted, Fearful, Sad:** The model has 0.00 for precision, recall, and F1-score for these classes, indicating it's not accurately identifying them.
- **Curious to dive deeper:** The model has a precision of 0.45, recall of 0.93, and F1-score of 0.61, suggesting it's relatively good at identifying this class, but with some false positives.

- **Happy, Neutral, Surprised:** The model has moderate precision (0.42-0.47) but low recall (0.07-0.09) for these classes, indicating it's missing many true instances while correctly identifying some.

Overall Performance:

- **Accuracy**: 0.45, indicating the model correctly classifies about 45% of instances overall.
- **Macro avg:** 0.44 for precision and recall, reflecting the average performance across classes without considering their distribution.
- **Weighted avg**: 0.35, considering the class distribution and giving more weight to classes with more instances.

Observations:

- The model struggles to identify certain classes (Angry, Disgusted, Fearful, Sad) accurately.
- It performs best for the "Curious to dive deeper" class, but with some false positives.
- Overall accuracy is moderate, suggesting potential for improvement.

3.1.2 Rnadomforest Classifier

Since our first algorithm encounters challenges with imbalanced data, I applied the class weight balancing technique to address the class imbalance problem.

	precision	recall	f1-score	support
Angry	0.00	0.00	0.00	162
Curious to dive deeper	0.48	0.90	0.63	16204
Disgusted	0.00	0.00	0.00	283
Fearful	0.07	0.01	0.02	218
Happy	0.52	0.14	0.22	5946
Neutral	0.46	0.22	0.30	8170
Sad	0.34	0.06	0.10	522
Surprised	0.49	0.15	0.23	6171
accuracy			0.48	37676
macro avg	0.30	0.18	0.19	37676
weighted avg	0.48	0.48	0.41	37676

Figure 4. Classification Report by using -Randomforest Classifier

Comparison in terms of Metrics:
Class-Specific Metrics Comparison:

- **Angry, Disgusted, Fearful, Sad:** Still 0.00 in both cases, indicating a persistent issue in identifying these classes.
- **Curious to dive deeper:**
  - Precision: 0.48 (Randomforest) vs. 0.45 (Naive Bayes)
  - Recall: 0.90 (Randomforest) vs. 0.93 (Naive Bayes)
  - F1-score: 0.63 (Randomforest) vs. 0.61 (Naive Bayes)
  - Slight improvements in precision and F1-score, but a slight decrease in recall in the Randomforest.
- **Happy, Neutral, Surprised:**
  - Generally higher precision and F1-scores in the image, suggesting better identification of positive instances.
  - Recall scores remain relatively low for these classes in both cases.

Observations :

- The Randomforest classifer show some overall improvement in accuracy and average performance.
- The model still struggles with certain classes (Angry, Disgusted, Fearful, Sad), highlighting a consistent issue.
- There are mixed results for the ”Curious to dive deeper” class, with trade-offs between precision and recall.
- The improvements in precision for Happy, Neutral, and Surprised classes suggest better identification of positive instances, but recall remains a challenge.

3.2 Deep Learning Approach

As our model continues to face challenges with imbalanced classes, I experimented with a transformer-based model called BERT-base-uncased to assess its performance.

3.2.1 Implementation

I employed the PyTorch framework to create a custom class, utilizing the DataLoader from PyTorch. Additionally, a pre-trained model from Hugging Face was invoked.Given that this is a multi-classification problem, it is necessary to specify the number of labels in the model.The CrossEntropyLoss function is employed as the loss function, given that the task involves multi-class classification.

Metrics Explanation:

	precision	recall	f1-score	support
Angry	0.00	0.00	0.00	162
Curious to dive deeper	0.68	0.73	0.70	16204
Disgusted	0.18	0.15	0.16	283
Fearful	0.41	0.21	0.28	218
Happy	0.46	0.44	0.45	5946
Neutral	0.51	0.56	0.54	8170
Sad	0.30	0.34	0.32	522
Surprised	0.52	0.40	0.45	6171
accuracy			0.58	37676
macro avg	0.38	0.35	0.36	37676
weighted avg	0.57	0.58	0.57	37676

Figure 5. Classification Report by using -BERT

Metrics Comparison

Overall Accuracy:

- BERT(Deep Learning) : 058
- Traditional Machine Learning : 0.48
- BERT has a higher overall accuracy, suggesting it’s correctly classifying more instances overall.

Class-Specific Metrics::

- Angry, Disgusted, Fearful, Sad: Both have 0.00 for all scores, indicating a persistent issue in identifying these classes with both approaches.
- Curious to dive deeper:
  - Precision: 0.68 (BERT) vs. 0.48 (ML)
  - Recall: 0.73 (BERT) vs. 0.98 (ML)
  - F1-score: 0.70 (BERT) vs. 0.63 (ML)
  - BERT has higher precision and F1-score, suggesting better positive identification, but lower recall, potentially missing some true instances.
- Happy, Neutral, Surprised:

  - Generally higher precision and F1-scores for BERT, suggesting better identification of positive instances.
  - Recall scores are mixed, with some higher and some lower for BERT.

Obervation :

- BERT shows promise with higher over all accuracy, precision and F1-scores for several classes, indicating better identification of positive instances.
- Both models struggle with certain classes (Angry, Disgusted, Fearful, Sad), highlighting a consistent challenge.
- The trade-offs between precision and recall for the ”Curious to dive deeper” class warrant further investigation.

Improvement Suggestions:

- The choice between BERT and ML approaches depends on the specific use case and priorities for precision, recall, and overall accuracy.
- Further exploration of hyperparameter tuning and data quality for both models could potentially improve performance.
- Understanding the reasons for the persistent issues with specific classes could guide model refinement and feature engineering efforts.
- Increasing the number of epochs could potentially enhance the model performance in BERT.

4 CODE

- **Task1_MLipynb**: Please refer this notebook for Machine Learning Approach.
- **Task1_DLipynb**: Please refer this notebook for Deep Learning Approach.

MY PORTFOLIO

Please feel free to explore my projects:

- Github