CIS 5300 Final Project Milestone 2

Gokul Nair, Rohan Saraogi, Shivani Prasad Bondapalli, Yash Agrawal

December 8, 2023

1 Evaluation Measure

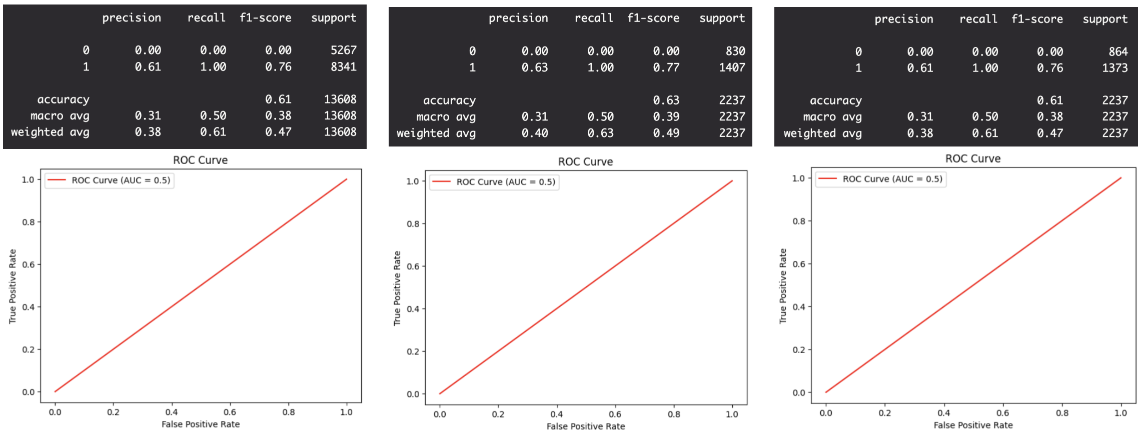
In our project, the final labels are 0/1 where 1 represents the tweets involving classification which are informative, and 0 is non-informative. Thus, we opt for the standard classification metrics like **accuracy**, **classification score** (**Precision, Recall, F1-Score**), and also display the **ROC curve/AUC score**. This is in accordance with the research papers we submitted in the literature review as well (`https://arxiv.org/abs/2004.11838`). Below is a formal definition, and explanation of how these measures work :

1. **Accuracy**: Ratio of the number of correct predictions to the total number of data samples.
2. **Precision**: Precision is calculated as the ratio of true positives to the sum of true positives and false positives. In other words, it's the percentage of positive identifications that were correct
3. **Recall**: Recall is calculated as the ratio of true positives to the sum of true positives and false negative. In other words, it measures how often a model is correct when predicting the target class.
4. **F1 score** - The F1 score is a single metric that combines precision and recall to assess the balance between true positive predictions and minimizing false positives and false negatives in classification tasks.
5. **ROC curve/AUC score** - A receiver operating characteristic (ROC) curve is a graphical representation of a model's performance by plotting the true positive rate against the false positive rate at various thresholds. The AUC score is the area under the curve.

2 Simple Baseline & Performance

The simple baseline that we have employed is a majority class classifier. This mainly identifies how the training data is distributed on the counts of the informative and non-informative tweets and uses this to predict new dev and test samples. We found that the majority classifier chose the informative label (1) as the most occurring label in the training set. Among the 13608 examples, 8341 were informative tweets, resulting in an accuracy of approximately 61% and a macro-F1 score of 0.38.

Using this classifier, we also evaluated it's performance on the dev and test data. This yielded an accuracy of 63% and a macro-F1 score of 0.39 on the dev data, and an accuracy of 61% and a macro-F1 score of 0.38 on the test data. The model performs poorly as the data distribution is the sole estimator of how the label for any future example is determined, without considering any other factors that would influence the prediction.



```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      5267
           1       0.61      1.00      0.76      8341

    accuracy                           0.61     13608
   macro avg       0.31      0.50      0.38     13608
weighted avg       0.38      0.61      0.47     13608
```

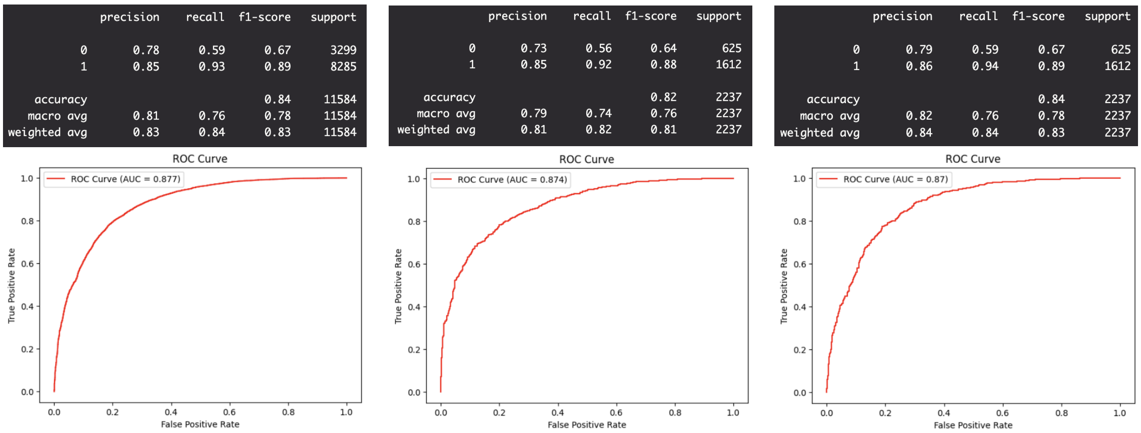
```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       830
           1       0.63      1.00      0.77      1407

    accuracy                           0.63      2237
   macro avg       0.31      0.50      0.39      2237
weighted avg       0.40      0.63      0.49      2237
```

```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       864
           1       0.61      1.00      0.76      1373

    accuracy                           0.61      2237
   macro avg       0.31      0.50      0.38      2237
weighted avg       0.38      0.61      0.47      2237
```

Figure 1: Simple baseline classification report and ROC curve for train (left), dev (middle), and test (right) data.

3 Strong Baseline & Performance

The sentence-transformer based text-only strong baseline model, leveraging the 'all-MiniLM-L6-v2' SentenceTransformer, establishes a robust foundation for multimodal disaster tweets classification. We generate embeddings for the tweet text and train a logistic regression classifier on it. The model utilized "label-text" as the target label for this task, and we performed deduplication based on tweet texts. This step was essential to handle cases where the same text corresponded to different image pairs, ensuring data integrity and preventing biases during training. The model achieved an accuracy of 83.5% and a macro-F1 score of 0.78 on the training set, demonstrating its proficiency in discerning disaster-related language. Generalizing to unseen data, it attained an accuracy of 82.1% and a macro-F1 score of 0.76 on the dev data, and an accuracy of 84.1% and a macro-F1 score of 0.78 on the test data. With relatively more balanced precision, recall, and F1-score metrics, the model is more effective in distinguishing between informative and non-informative tweets. This baseline not only serves as a benchmark but also suggests potential avenues for future improvement, such as fine-tuning strategies or exploring alternative transformer architectures.



```
              precision    recall  f1-score   support

           0       0.78      0.59      0.67      3299
           1       0.85      0.93      0.89      8285

    accuracy                           0.84     11584
   macro avg       0.81      0.76      0.78     11584
weighted avg       0.83      0.84      0.83     11584
```

```
              precision    recall  f1-score   support

           0       0.73      0.56      0.64       625
           1       0.85      0.92      0.88      1612

    accuracy                           0.82      2237
   macro avg       0.79      0.74      0.76      2237
weighted avg       0.81      0.82      0.81      2237
```

```
              precision    recall  f1-score   support

           0       0.79      0.59      0.67       625
           1       0.86      0.94      0.89      1612

    accuracy                           0.84      2237
   macro avg       0.82      0.76      0.78      2237
weighted avg       0.84      0.84      0.83      2237
```

Figure 2: Strong baseline classification report and ROC curve for train (left), dev (middle), and test (right) data.