Tytus Felbor
Prof. Hu
CPSC444 : Final Project Report

Essay Classification Problem: Is it student-written or LLM-generated?

---

1. Problem Description

At the heart of the problem lies the classification of essays as either student-written (label 0) or LLM-generated (label 1) using a binary classification model. The dataset used to train the model (`train_essays.csv`), includes 1,378 essays, with a significant imbalance of classes. The uneven distribution of classes in our training dataset, stems from 1,375 of all the essay entries being student-written, and only 3 representing the LLM-generated body/class.

With that being said, our challenges are the following:

- Severe Class Imbalance - only around 0.2% of essays are LLM-generated, risking the model being prone to bias in favor of choosing the student-written class, the majority class.
- Text-Based Input - our data (the essays) are of varied size (different number of strings composes each essay), which forces us to extensively process our data, so features of each can be extracted and its format is standardized. That way we ensure that the crucial linguistic properties are preserved, and the size of our input is consistent, so that it can be easily fed into our model with no issues.
- Model Optimization: achieved through comparing different configurations of hyperparameters and levels of classification thresholds against the results they yield, in order to ultimately find the most optimal solution. Specifically, one which yields the best results while working with the limited computational resources of Google Colab's free tier.

---

2. ANN Architecture

The architecture of our artificial neural network is as follows:

- Input Layer - 100 nodes, each 1 dimensional embedding vector holds 100 distinct features (averaged GloVe 100D embeddings per each essay).
- Hidden Layer - 64 nodes, ReLU activation function to capture the non-linear & complex relationships/patterns in our data.
- Dropout - the value of 0.2 should help with preventing the phenomena of overfitting the oversampled data.
- Output Layer - 1 node using sigmoid activation function, it outputs the probability of an essay being LLM-generated.
- Parameters - 6,529 (6,464 for our hidden layer, 65 for the output), compact to work with our limited Colab vers.
- Loss - Binary cross-entropy, standard for binary classification.
- Optimizer - Adam, since it's known to be efficient in text processing.

---

3. Parameters Tuned

Through manual hyperparameter tuning with early stopping our model achieved optimal configuration, displayed as follows:

- **Varied Hyperparameter Test Cases**
  - Learning Rate - `[0.001, 0.0001]`
  - Batch Size - `[32, 64]`
  - Dropout Rate - `[0.2, 0.3]`

- **Different Combinations (4 in total)**
    - `learning_rate=0.001, batch_size=32, dropout_rate=0.2`
      `F1-Score: 0.991, Accuracy: 0.995`
    - `learning_rate=0.0001, batch_size=32, dropout_rate=0.2`
      `F1-Score: 0.822, Accuracy: 0.910`
    - `learning_rate=0.001, batch_size=64, dropout_rate=0.2`
      `F1-Score: 0.979, Accuracy: 0.987`
    - `learning_rate=0.001, batch_size=32, dropout_rate=0.3`
      `F1-Score: 0.991, Accuracy: 0.995`
- **Best Hyperparameters** - highest `F1-Score` of `0.991` & `Accuracy` of `0.995`
    - `learning_rate=0.001, batch_size=32, dropout_rate=0.2`
      &
    - `learning_rate=0.001, batch_size=32, dropout_rate=0.3`
- **Early Stopping**
    - the model stops if the current epoch's validation loss didn't improve for the past 3 epochs (with a max of 20 epochs), aiming at contributing to overall efficiency.
- **Threshold Optimization**
    - adjusting the classification threshold from `0.5` to `0.7`, resulted in the improvement of the `F1-Score` to `0.996` and the `Accuracy` to `0.995`

---

4. Results

The model delivers the following results:

- **Best Model Performance** (testing with a validation set of 391 samples)
    - `F1-Score`
        - `0.996` (using threshold of `0.7`, an improvement from `0.991` at threshold levels of `0.5`).

    - `Accuracy`
        - `0.997` (up from `0.995` when using a threshold level of `0.5`).
- **Best Hyperparameters**
    - `learning_rate=0.001, batch_size=32, dropout_rate=0.2`

      &
    - `learning_rate=0.001, batch_size=32, dropout_rate=0.3`

---

5. Improvement Strategies

In order to enhance the performance of our model we'd need to implement the following changes:

- in terms of our data it would prove useful to add more LLM-generated essays into our dataset to improve its diversity and class balance.
- it could also prove valuable to increase the size of our hidden layer (128 nodes would be enough) or perhaps add a second layer so more complex patterns can be captured by our model.
- we could also test more hyperparameter combinations to find one that outperforms our solution.

To conclude, this ANN model quite effectively performs the binary classification task, achieving a validation F1-Score of 0.996 and Accuracy of 0.997 after a few optimizations. The pipeline of our data, starting with GloVe embeddings, proceeding with random oversampling, and culminating with manual tuning, was a basic, yet efficient approach, which aligned with Google Colab's computational constraints. The small, low diverse dataset and severe class imbalance however, suggest the credibility of the test set and its performance might be questionable, and that might be another reason why our result looks so promising.