Classical Baseline Implementation Using TF-IDF and Naive Bayes

As part of our team's project on semantic classification of math problems, my responsibility was to establish a robust classical baseline using traditional machine learning methods. This baseline acts as a foundation for later comparison with more advanced transformer-based models.

Objectives:

- Preprocess and clean the dataset for basic NLP modeling.
- Extract relevant textual features using TF-IDF vectorization.
- Train and evaluate a simple yet effective classifier to benchmark initial performance.
- Save predictions for the test set for comparison and submission.

Methodology:

1. **Data Loading & Cleaning**
   I implemented a function to load training and test datasets and applied basic preprocessing. This included:
   * Lowercasing text
   * Replacing math expressions with a MATH placeholder
   * Removing non-alphanumeric characters and normalizing whitespace

2. **Feature Extraction**
   I used the TfidfVectorizer with unigrams and bigrams and limited the vocabulary to 5000 features. This transformed the cleaned text into a sparse matrix suitable for classification.

3. **Model Selection & Evaluation**
   For classification, I chose the Multinomial Naive Bayes model from Scikit-learn, which is well-suited for TF-IDF feature vectors. I used an 80-20 train-validation split with stratification to maintain class balance. Evaluation was done using the macro-averaged F1-score, which is essential for our multi-class task with uneven class distribution.

4. **Submission Preparation**
   Finally, I retrained the model on the entire training dataset and generated predictions on the test set. The results were saved as submission_nb.csv.

Results:

The model achieved a respectable macro-F1 score on the validation set. While we expected limitations in semantic depth with traditional models, this baseline provides a meaningful reference point before advancing to transformer models.