# Model1：Bert-CNN

**Justification and Explanation:**

In this project, we use a combination of the BERT model and Convolutional Neural Network (CNN) for hate speech detection. The reasons for selecting this combined model are:

1. **BERT Model:** BERT (Bidirectional Encoder Representations from Transformers) has shown excellent performance in natural language processing tasks, particularly in understanding context and capturing semantic information.
2. **Convolutional Neural Network (CNN):** Adding CNN layers on top of the BERT model can further capture feature patterns. Pooling layers and fully connected layers help in classification, potentially improving model performance.

**Experimental Setup**

**Model Parameters:**

* **Learning Rate:** 5e-5
* **Batch Size:** 16
* **Epochs:** 3
* **Loss Function:** Negative Log-Likelihood Loss (NLLLoss)

**Evaluation Metrics:**

* **Precision**
* **Recall**
* **F1 Score**
* **Classification Report**
* **Accuracy**

These metrics provide a comprehensive evaluation of the model's performance, especially when dealing with imbalanced datasets.

**Data Split:**

* The dataset is split into training (80%), validation (10%), and test sets (10%) to ensure a thorough evaluation of the model's performance.

**Main Findings and Results:**

1. **Model Performance:**
   * During training and validation, the model's loss gradually decreases, indicating that the model is learning and optimizing.
   * The final classification report on the test set shows the model's performance across different classes, and the accuracy provides an overall performance evaluation.
2. **Impact of Class Imbalance:**
   * The results show variations in precision and recall for different classes, with the model potentially performing worse on minority classes.
3. **Comparison to Other Methods:**
   * Although not directly compared to state-of-the-art (SOTA) methods, the performance of this model suggests that with appropriate model selection and data preprocessing, good results can be achieved in hate speech detection tasks.