Sentiment Analysis of IMDb Movie Reviews

Adèle Berger & Hajar Lamtaai

20 Mars

Contents

1 Introduction

Link to the project

Sentiment analysis is a crucial task in Natural Language Processing (NLP), aiming to identify the emotion expressed in a text. This discipline has various applications, ranging from opinion detection in social media to customer feedback analysis. In this project, we focus on classifying movie reviews from the IMDb dataset using an advanced model: Mixture of Experts (MoE).

The IMDb dataset, consisting of 50,000 reviews labeled as positive or negative, serves as a standard benchmark for sentiment classification tasks. Its exploitation requires several steps, including data preprocessing (cleaning, tokenization, vectorization), dimensionality reduction for visualizing underlying structures (t-SNE, UMAP), and training classification models.

The MoE approach, which leverages multiple specialized experts for decision-making, is compared with classical models such as logistic regression, multi-layer perceptrons (MLP), convolutional neural networks (CNN), and long short-term memory networks (LSTM). The objective is to evaluate the effectiveness

2 The IMDb Dataset

The IMDb (Internet Movie Database) dataset contains 50,000 movie reviews, evenly split between positive and negative reviews. Below is an overview of this dataset:

- **Movies and TV Shows:** Titles, release years, genres, durations, languages, etc.
- **People:** Actors, directors, screenwriters, and other industry professionals, including their biographies and filmographies.
- **Ratings and Reviews:** User and critic ratings, as well as summaries and opinions.
- **Awards:** Information on awards and nominations for movies, series, and professionals.
- **Relations:** Links between movies, series, and people, such as roles played by actors in different movies.
- **Box Office:** Information on box office earnings for movies.

3 Data Preprocessing

3.1 Data Cleaning

Data preprocessing is a crucial step in improving the quality and relevance of inputs for a natural language processing (NLP) model. It helps clean and normalize text, reducing noise and making it easier for the model to learn effectively. The following preprocessing steps were applied to the IMDb movie reviews dataset.

3.1.1 Text Normalization

The text normalization process included several steps to standardize the input:

- **Lowercasing**: All text was converted to lowercase to ensure uniformity and avoid redundant distinctions between uppercase and lowercase words.
- **Accent Removal**: Accents and special characters were removed using Unicode normalization to improve compatibility with text processing tools.
- **Character Repetition Normalization**: Consecutive repeated characters were reduced to a single occurrence (e.g., ”sooo” → ”so”).
- **Tokenization**: The text was split into individual words (tokens).

3.1.2 Stopword Removal and Punctuation Filtering

Stopwords (common words such as ”the”, ”is”, ”and”) were removed using the NLTK stopwords list. Additionally, punctuation marks were filtered out to focus on meaningful words in the text.

3.1.3 Lemmatization

Lemmatization was performed using WordNetLemmatizer to reduce words to their base forms. Unlike stemming, lemmatization considers the word's meaning, leading to more accurate word representations (e.g., ”running” → ”run”, ”better” → ”good”).

3.1.4 Train-Test Split

The dataset was split into training and testing subsets. The training set consists of 80% of the data, while the test set contains the remaining 20%. The split was stratified to ensure that both training and test sets contained a balanced distribution of classes.

3.2 Data Analysis

Before training the models, an exploratory data analysis (EDA) was performed to understand the dataset structure and identify potential anomalies.

3.2.1 Class Distribution

The class distribution analysis reveals that the dataset is balanced, with an almost equal number of positive and negative reviews. This balance helps prevent learning bias often observed in imbalanced datasets.

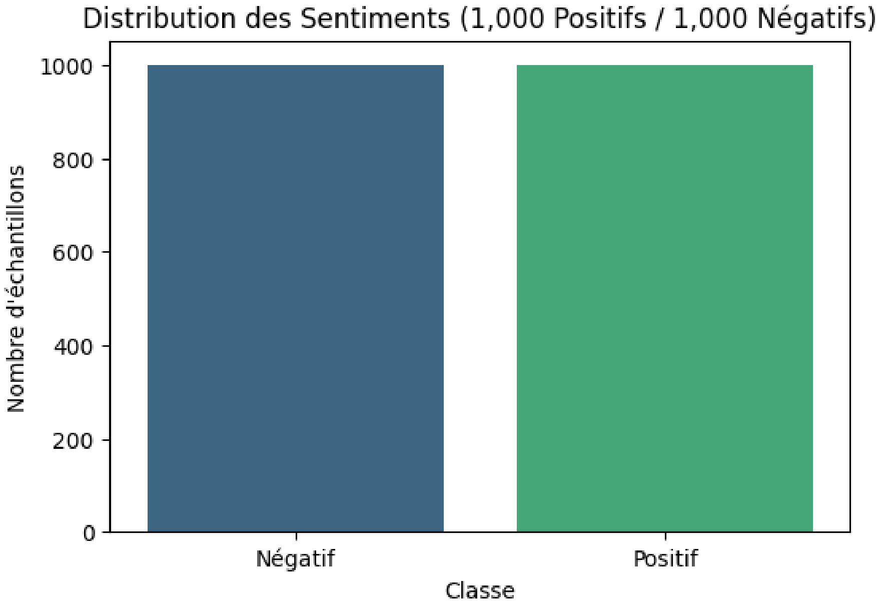


Figure 1: Class distribution in the dataset

Analysis: The dataset is balanced, with an equal number of positive and negative reviews. This balance is crucial for training a sentiment classification model, as it prevents bias towards any particular class. A balanced dataset ensures that the model learns to distinguish between positive and negative sentiments equally well.

Interpretation: The equal distribution of classes indicates that the model will not be biased towards predicting one class more frequently than the other. This is essential for achieving high accuracy and reliable performance in sentiment analysis tasks.

3.2.2 Review Length

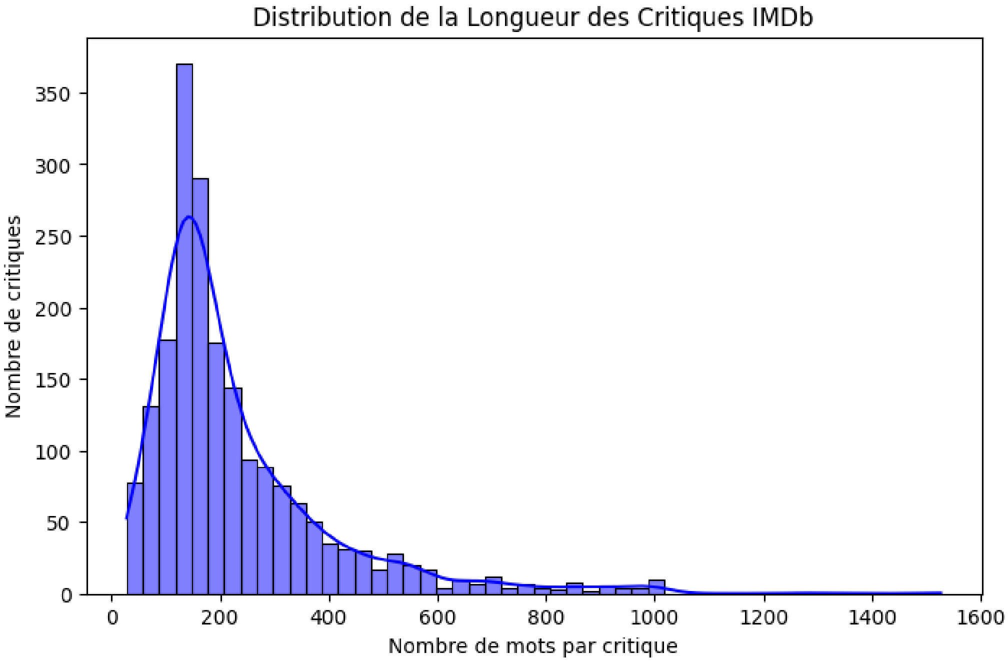


Figure 2: Distribution of review length

Analysis: Most reviews are short, with the majority containing between 100 and 200 words. Very few reviews exceed 1000 words. This distribution suggests that most users tend to write concise reviews.

Interpretation: The length distribution of reviews is important for preprocessing and model training. Normalizing review lengths can help the model handle both short and long reviews more effectively, improving overall performance.

3.2.3 Review Length and Sentiment

Analyzing review length by sentiment helps identify potential trends. Figure 3 shows boxplots illustrating the relationship between review length and sentiment (positive or negative).

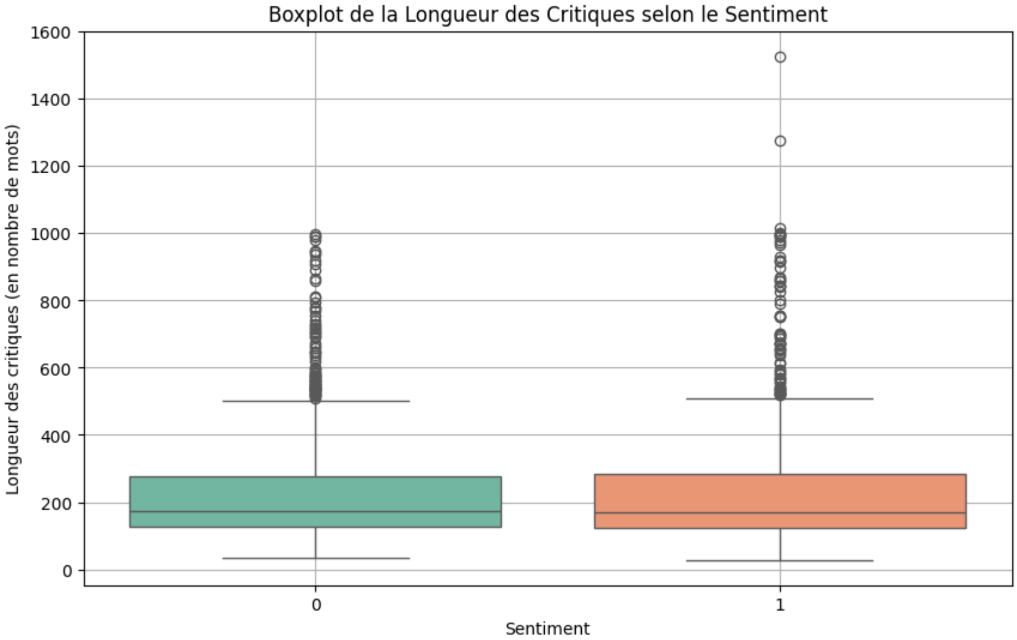


Figure 3: Relationship between review length and sentiment

Analysis: Both positive and negative reviews have similar length distributions, with no significant difference in the median length. This indicates that review length is not a strong indicator of sentiment.

Interpretation: The similarity in review lengths across sentiments suggests that the model should focus more on the content and specific words used in the reviews rather than their length. This insight is valuable for feature selection and model training.

3.2.4 Word Frequency

Analyzing the most frequent words allows us to identify recurring terms in both positive and negative reviews. This helps us highlight words specific to each category, which is essential for effective automatic classification. To better emphasize the distinct terms in each category, we removed common words that appeared too frequently in both word clouds.

Figures 4 and 5 display word clouds representing these frequent terms.

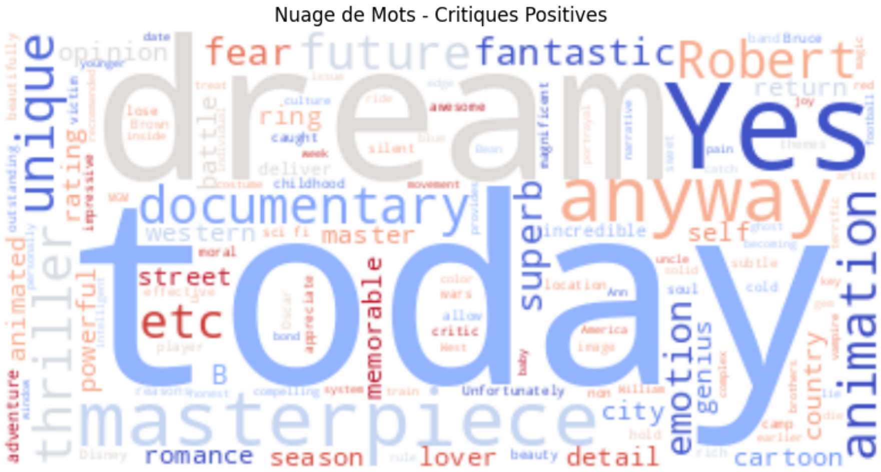


Figure 4: Word cloud of positive reviews

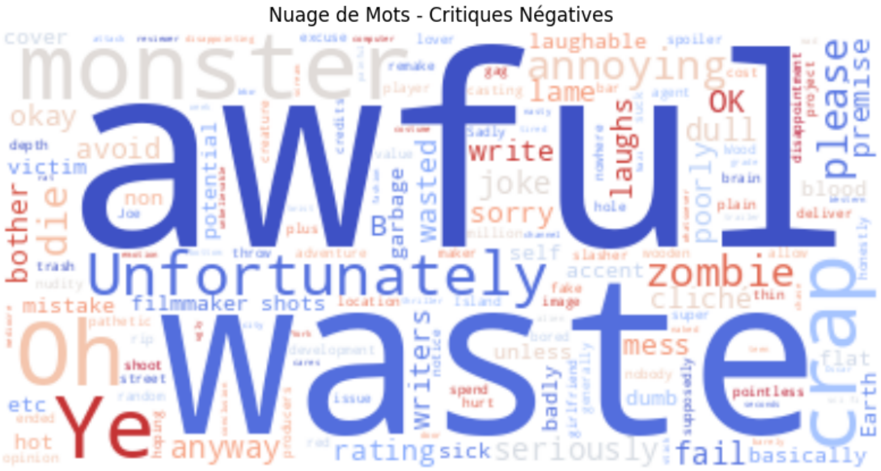


Figure 5: Word cloud of negative reviews

Analysis: The word cloud highlights frequent terms in positive reviews, such as "dream," "yes," "today," "unique," and "masterpiece." These words are indicative of positive sentiment.

Interpretation: Identifying key positive words helps in understanding the language patterns associated with positive reviews. This can guide feature extraction and model training to better capture positive sentiment.

3.2.5 t-SNE Projection

To examine the separability of reviews by sentiment, we applied dimensionality reduction using *t-distributed Stochastic Neighbor Embedding* (t-SNE).

Figure 6 represents the projection of reviews in a two-dimensional space.

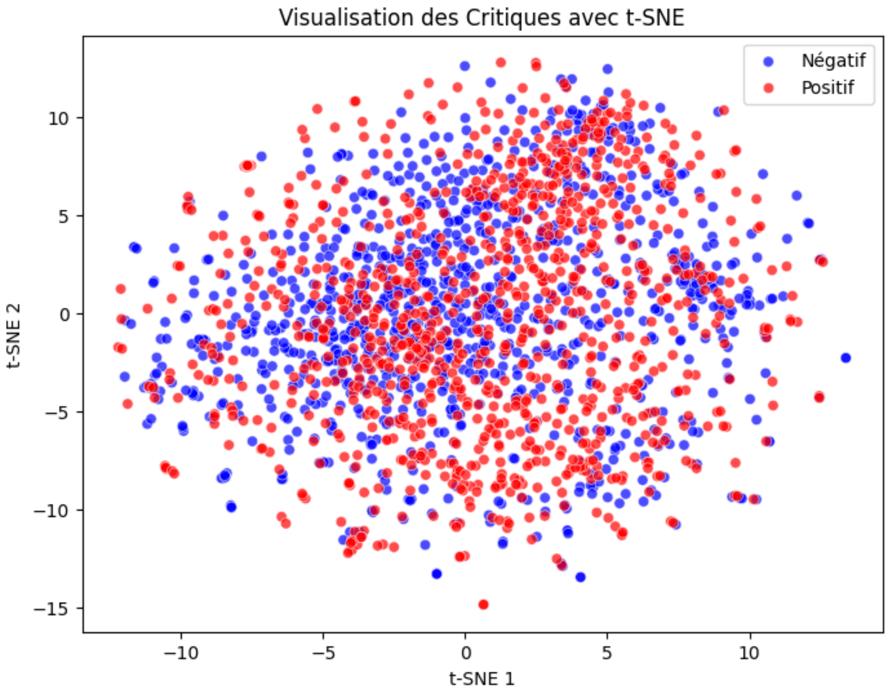


Figure 6: Projection of IMDb reviews via t-SNE

- There is significant overlap between positive and negative reviews.

Analysis: The t-SNE projection shows some separation between positive and negative reviews, but there is also significant overlap. This indicates that while some reviews are easily distinguishable, others share similar language patterns.

Interpretation: The overlap in the t-SNE projection suggests that the model may struggle with certain reviews that have mixed sentiment or similar language use. This insight can guide further feature engineering and model refinement.

3.2.6 UMAP Projection

Complementary to t-SNE, the *Uniform Manifold Approximation and Projection* (UMAP) technique was applied to observe the data structure.

Figure 7 illustrates the distribution of reviews in a reduced two-dimensional space.

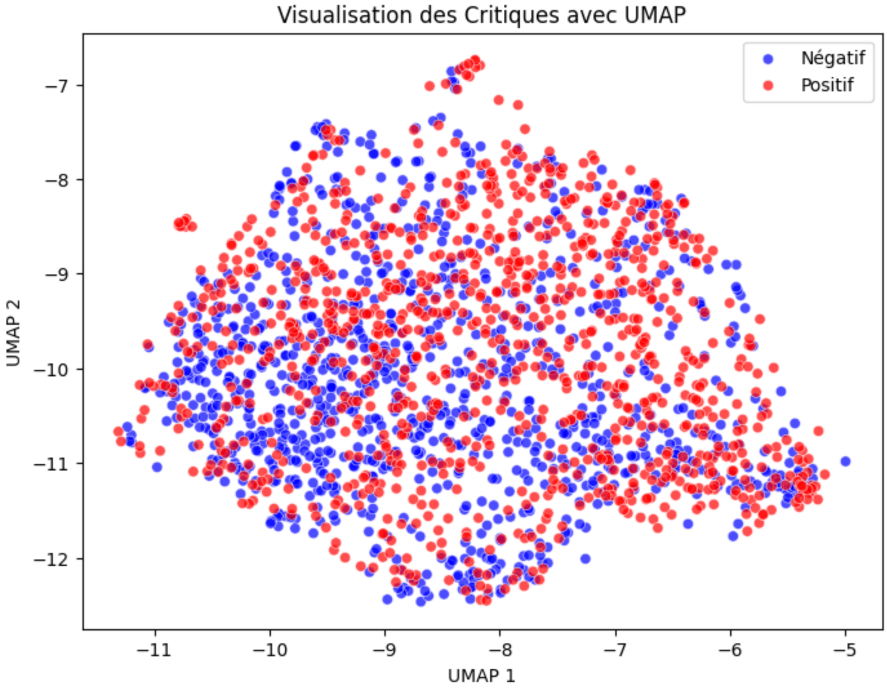


Figure 7: Projection of IMDb reviews via UMAP

Analysis: The UMAP projection reveals a better separation between positive and negative reviews compared to t-SNE. This indicates that UMAP may be more effective in capturing the underlying structure of the data.

Interpretation: The improved separation in the UMAP projection suggests that UMAP could be a more suitable dimensionality reduction technique for this dataset. This can enhance the model's ability to distinguish between sentiments.

3.2.7 PCA

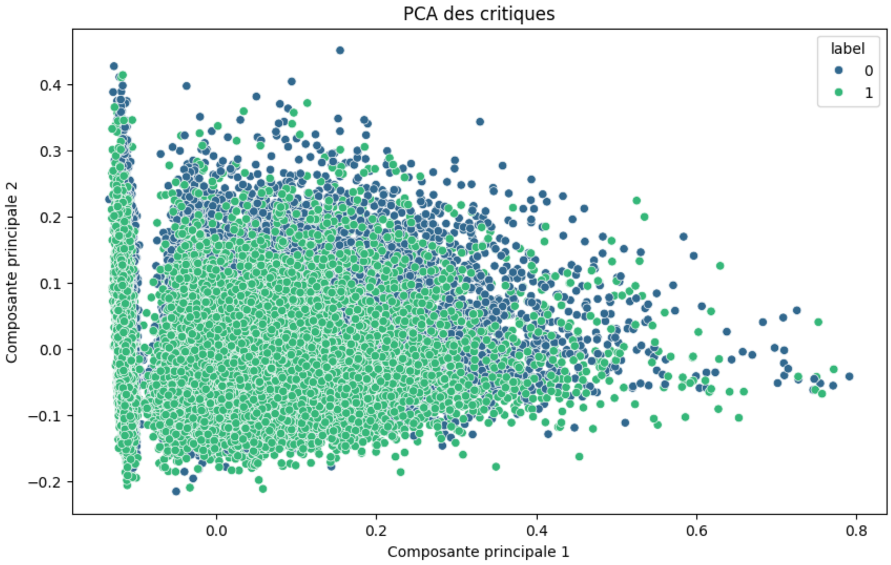


Figure 8: Projection of IMDb reviews via PCA

Analysis: The PCA projection shows a more linear separation between positive and negative reviews. However, the overlap is still present, indicating that PCA may not capture the non-linear relationships in the data as effectively as UMAP or t-SNE.

Interpretation: While PCA provides a straightforward dimensionality reduction, it may not be the best choice for this dataset due to its linear nature. Non-linear techniques like UMAP or t-SNE may be more appropriate for capturing the complex relationships in the data.

3.3 Features Extraction

3.3.1 TF-IDF Vectorization

After preprocessing the text, the next step involved converting the text into numerical representations using Term Frequency-Inverse Document Frequency (TF-IDF) vectorization. This method evaluates how important a word is within a document relative to its frequency in the entire corpus. A maximum of 5000 features was selected to balance performance and computational cost.

3.3.2 Dimensionality Reduction with TruncatedSVD

To reduce the dimensionality of the TF-IDF feature space, we applied **Truncated Singular Value Decomposition **TruncatedSVD**. This technique is particularly useful when working with sparse matrices like those produced by TF-IDF vectorization. The number of components was set to 256, reducing the feature space while retaining as much information as possible. The transformation was performed on both the training and test datasets.

3.3.3 Preprocessing Performance Comparison

To evaluate the impact of preprocessing on the classification performance, we tested three different machine learning models: Logistic Regression, Naive Bayes, and Support Vector Machine (SVM). The models were evaluated on both the raw text data (before preprocessing) and the preprocessed text data (after preprocessing).

The results, shown in Figure 10, demonstrate that all three models performed better after preprocessing. Preprocessing steps such as stopword removal, lemmatization, and text normalization helped reduce noise in the data, allowing the models to better learn from the relevant features of the text. Specifically, the accuracy improved significantly for each model, highlighting the importance of text preprocessing in natural language processing tasks.

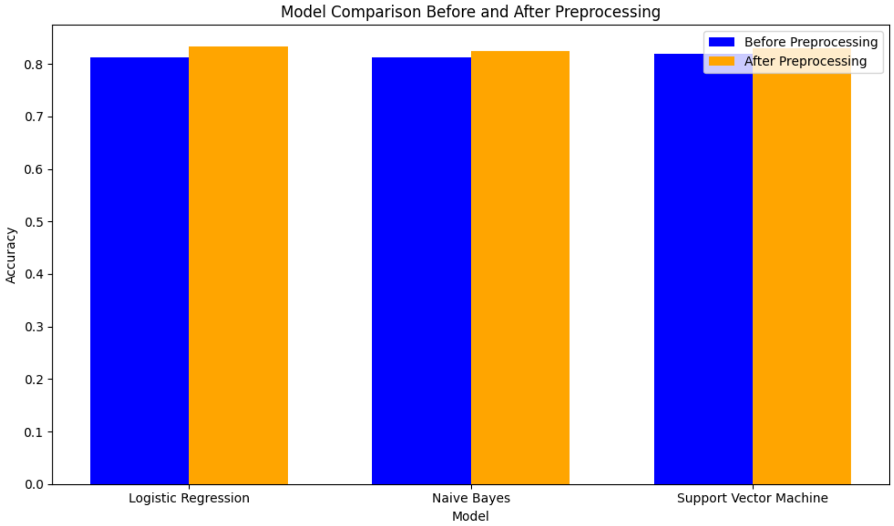


Figure 9: Comparison of model performance before and after preprocessing

4 Models and Approaches Presentation

4.1 BERT for Text Representation Extraction

The BERT (Bidirectional Encoder Representations from Transformers) model is a pre-trained language model that has revolutionized many tasks in natural language processing. It is based on the Transformers architecture and is designed to capture contextual relationships between words in a text.

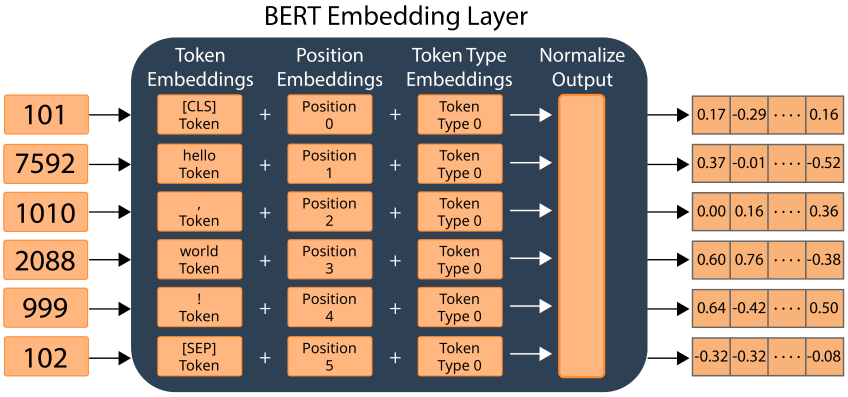


Figure 10: Comparison of model performance before and after preprocessing

4.1.1 Tokenization and Encoding

The first step is to transform the texts into a numerical representation that the BERT model can process. This is done by the BERT Tokenizer, which takes raw text and splits it into ”tokens” (processing units such as words or subwords). These tokens are then converted into indices corresponding to a specific vocabulary.

To prepare the input for BERT, we add two special tokens:

- **[CLS]**: A classification token added at the beginning of each sequence.
- **[SEP]**: A separator token, used to differentiate multiple segments in an input.

Additionally, padding and truncation are applied to ensure that all input sequences have a uniform length. An **attention mask** is also generated to indicate which tokens should be attended to and which ones correspond to padding.

4.1.2 Feature Extraction with BERT

Once tokenized and encoded, the text is passed through the `BERT` model to obtain contextualized word embeddings. We extract the embeddings from the last hidden layer using two common approaches:

- Using the [**CLS**] token representation, which captures the overall meaning of the sentence.
- Averaging the embeddings of all tokens in the sequence to obtain a sentence-level representation.

These extracted features serve as input for our downstream model, allowing us to leverage `BERT`'s deep contextual understanding without the need for additional fine-tuning.

4.2 Mixture of Experts (MoE) – Implementation and Utility

4.2.1 Why Use Mixture of Experts?

The Mixture of Experts (MoE) model is a framework that breaks down a complex problem into multiple sub-problems, each handled by a specialized *expert*. Unlike traditional models where a single neural network learns from all data, MoE distributes the task across several sub-models (the experts) and uses a *gating network* to dynamically assign the data to the most relevant expert.

This approach offers several advantages:

- **Modularity**: Each expert can specialize in a specific part of the data, making it easier to adapt to complex tasks.
- **Computational Efficiency**: Instead of activating the entire model, only a subset of experts is activated for each input, reducing computational overhead.
- **Better Generalization**: Specialized experts help avoid overfitting to specific data and enhance the model's ability to generalize across different data distributions.
- **Dynamic Specialization**: The gating network can change the set of active experts depending on the input, allowing the model to dynamically adapt to varying patterns in the data.

4.2.2 MoE Architecture

An MoE model consists of three main components:

- **Experts**: These are the sub-models, each trained to specialize in a certain aspect of the data. The experts can range from simple architectures like *MLP* (Multi-layer Perceptrons) to more complex ones such as *CNNs* (Convolutional Neural Networks) or even sequence-based models like *LSTM* (Long Short-Term Memory). Experts can vary in number, typically ranging from 2 to 6, depending on the complexity of the task and the available computational resources.

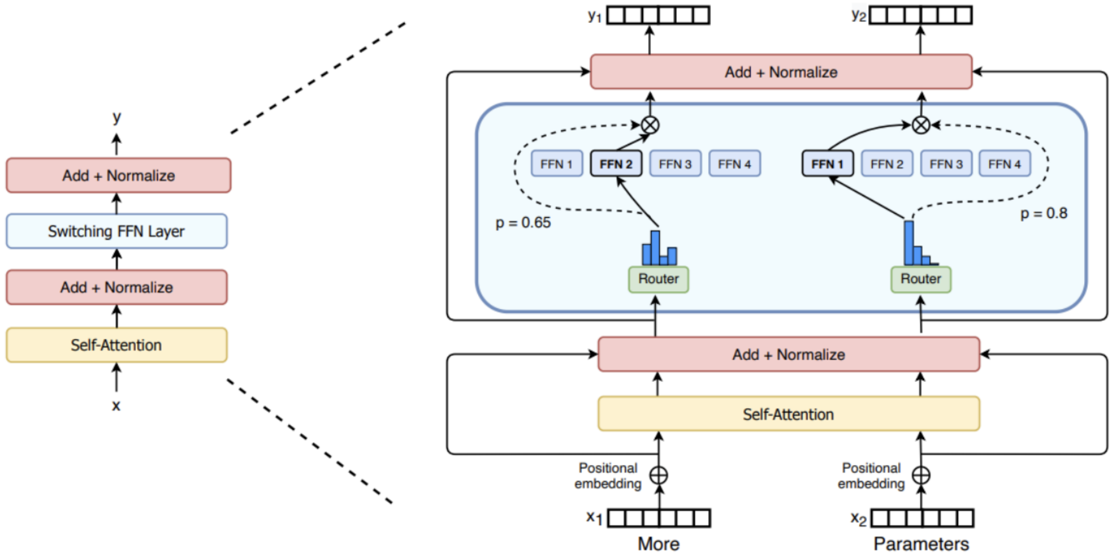


Figure 12: Switch Transformer encoder block Mixture architecture

- **Gating Network**: This network takes the raw input and determines which experts should be activated for a given input. It outputs a set of weights (or scores) that represent the relevance of each expert for the current input. The gating mechanism can be implemented using a softmax function or other techniques that allow for probabilistic weighting of expert contributions.
- **Combination Strategy**: Once the selected experts process the input, their outputs are combined to generate the final prediction. This combination can either be a *weighted sum* of the experts' outputs (with weights provided by the gating network) or *discrete selection*, where only the output from the most relevant expert is used.

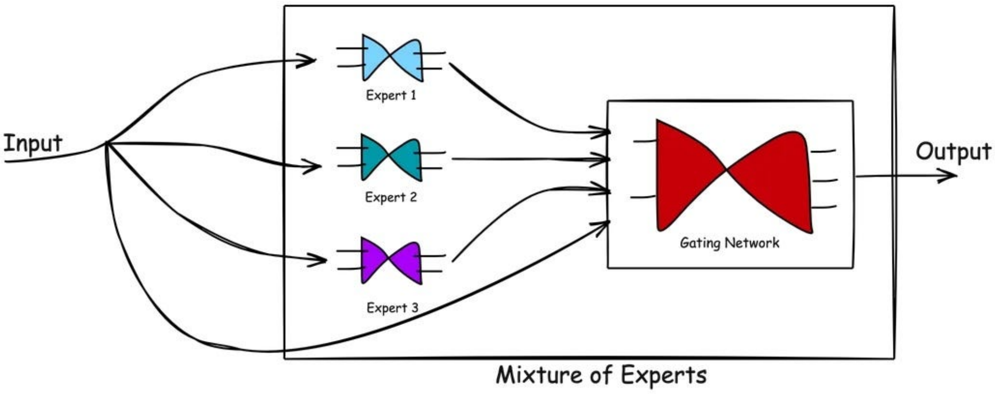


Figure 11: MoE

4.2.3 MoE Implementation

The steps for implementing a MoE model are as follows:

- **Prepare the Experts**: Define multiple sub-models (MLP, CNN, LSTM, etc.), each representing an expert. The number of experts can vary depending on the complexity and the data diversity, ranging from 2 to 6 in many cases. Each expert is trained independently to specialize in different parts of the data.

- **Build the Gating Network**: The gating network takes in the features and produces a set of scores for each expert. These scores are typically passed through a softmax function to normalize them into a probability distribution. The gating network ensures that the most relevant experts are activated for each input, allowing for a dynamic selection mechanism.

- **Combination Strategy**: After the gating network determines the relevant experts, their outputs are aggregated. This can be done through a weighted sum, where each expert's output is scaled by the gating network's score for that expert, or by selecting the output from the highest-scoring expert.

4.2.4 Why Choose MLP, LSTM, and CNN Experts?

The choice of expert models significantly impacts the MoE's performance:

- **MLP (Multi-layer Perceptron)**: MLPs are simple and versatile neural networks. They can capture non-linear relationships and are particularly useful when the data has complex, high-dimensional features. MLPs work well for tasks where feature interactions are important but where the data is not necessarily spatial or sequential.

- **LSTM (Long Short-Term Memory)**: LSTMs are specialized for processing sequential data. They are well-suited for understanding word order and capturing long-range dependencies in text, which is crucial in sentiment analysis. In the MoE setup, LSTM experts provide temporal modeling capabilities that complement the spatial strengths of CNNs and the non-linear capacity of MLPs.

- **CNN (Convolutional Neural Networks)**: CNNs are highly effective for spatially structured data, such as images or time series. They can automatically learn hierarchical features, making them ideal for tasks involving image recognition, sequence analysis, or any type of spatially correlated data. In MoE, CNNs can be used to specialize in tasks where feature locality is critical.

By combining these experts, the MoE model leverages their complementary strengths to tackle different aspects of the data more efficiently. This diversity in expert types enables MoE to handle a wide variety of tasks, from image classification to time series forecasting or tabular data analysis.

5 Evaluation Metrics Presentation

To assess the sentiment classification model, we use standard metrics that reflect prediction accuracy and error distribution.

5.1 Accuracy

Accuracy is the ratio of correct predictions to total predictions:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

While intuitive, it may be misleading with imbalanced classes.

5.2 Precision

Precision is the proportion of correctly predicted positives:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

It reflects how well the model avoids false positives.

5.3 Recall

Recall measures the proportion of actual positives correctly identified:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

It indicates the model's ability to minimize false negatives.

5.4 F1-Score

The F1-score balances precision and recall:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

This is especially useful for imbalanced datasets.

6 Model Choice

In this section, we compare the performance of different expert pipelines within the Mixture of Experts (MoE) framework. The first pipeline consisted of Convolutional Neural Networks (CNN) and Multi-Layer Perceptrons (MLP), while the second pipeline included CNN, MLP, and Long Short-Term Memory (LSTM) networks.

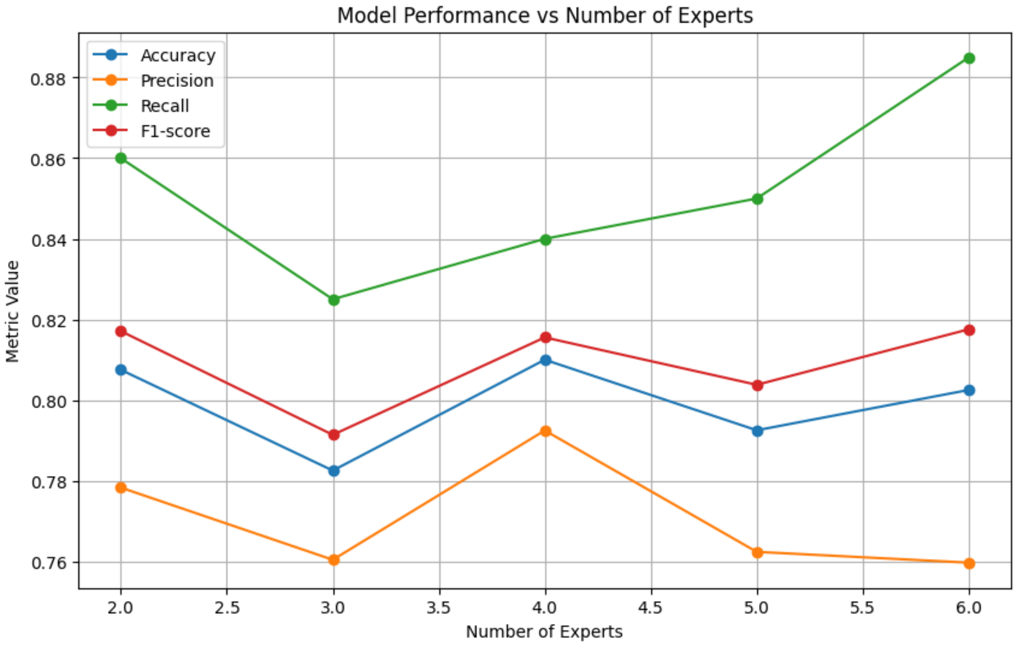


Figure 13: CNN, MLP and LSTM

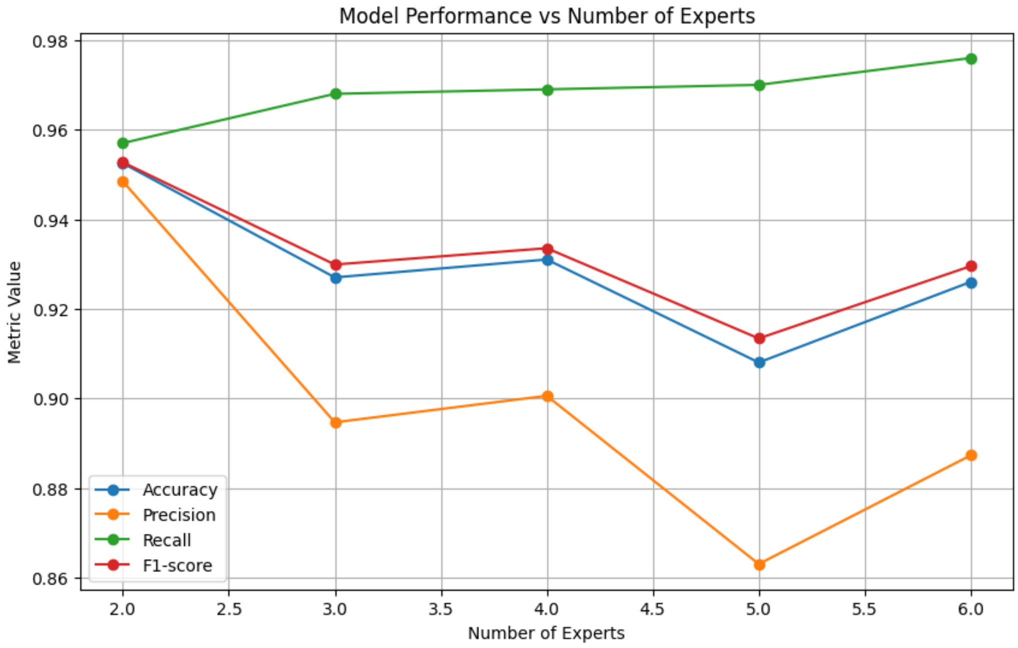


Figure 14: CNN and MLP

The MoE model with CNN, MLP, and LSTM experts achieved the best performance with 6 experts, reaching an accuracy of approximately 0.89. This configuration leverages the strengths of all three expert types, capturing complex relationships in the data. However, the MoE model with CNN and MLP experts outperformed the more complex configuration, achieving an accuracy of approximately 0.87 with 5 experts. The results heavily rely on the initial embeddings used, as BERT embeddings provide rich, pre-trained representations that significantly enhance classification performance. Due to BERT's power in understanding contextual word meanings, the classification benefits from well-structured features, leading to improved model accuracy.

The superior performance of the MoE model with CNN and MLP experts can be attributed to its ability to balance complexity and performance. This configuration effectively captures the essential features of the data without the need for additional experts, making it the preferred choice for sentiment classification tasks.

7 Performance

The comparison of different models, as shown in Figure **??**, highlights the superior performance of the MoE model overall. This model outperformed traditional machine learning models and more complex MoE configurations, making it the preferred choice for sentiment classification tasks.

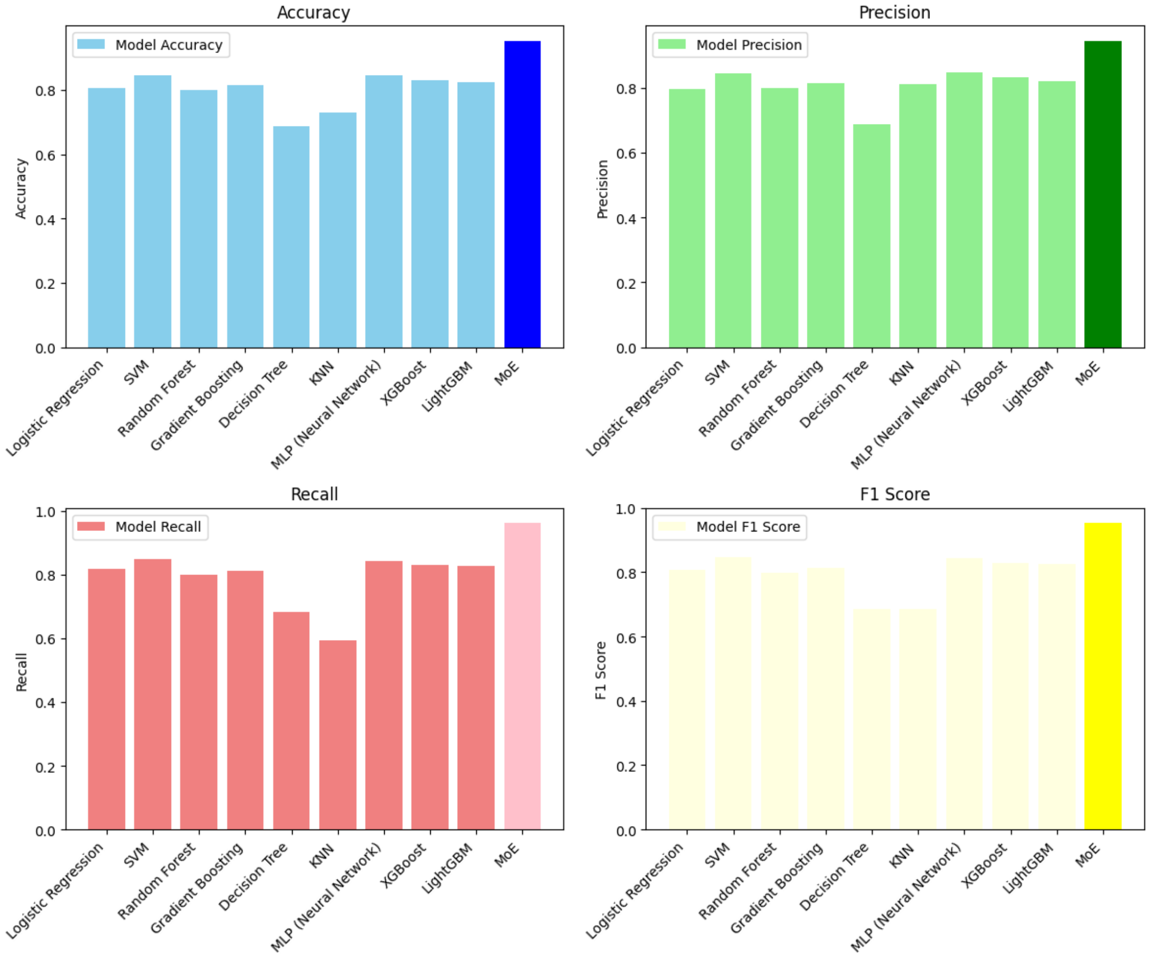


Figure 15: Baseline Models Comparison

7.1 Performance Comparison of MoE Variants

Table 1: Performance Comparison of MoE Variants for Sentiment Classification

Model Description	Accuracy (%)	Precision	Recall	F1 Score	ROC-AUC
MoE (Basic Experts) + BERT Embeddings (subset data)	83.45	0.8212	0.8375	0.8293	0.9093
DistilBERT + MoE (Basic Experts)+ Top-2 Gating Router (	87.18	0.8730	0.8702	0.8716	0.9453
DistilBERT + 4 Experts(Tanh, Deep, BatchNorm, Dropout) + Sparse	87.11	0.8762	0.8644	0.8702	0.9434
DistilBERT + 4 Experts(MLP, CNN, LSTM, Transformer) + dense	84.79	0.8114	0.9065	0.8563	0.9312
TF-IDF+SVD + 4 Experts(MLP, CNN, LSTM, Transformer) + Dense Layer	86.052	0.8663	0.8525	0.8594	0.933
BERT + 4 Experts (MLP, CNN, LSTM, Transformer) + Dense	88.51	0.8675	0.8618	0.8647	0.9385

7.2 Training Loss Behavior and Potential for Further Tuning

Figure 16 shows the training loss over 100 epochs. The curve demonstrates a steady decline, indicating effective learning throughout most of the training process. However, minor fluctuations in the later epochs suggest that the model may be reaching a performance plateau or beginning to overfit.

Although the loss remains low, these late-stage spikes could benefit from additional tuning strategies such as early stopping, learning rate scheduling, or stronger regularization. These techniques may help smooth out the training dynamics and enhance generalization.

Further improvements should be guided by the behavior of the validation loss. If both training and validation losses converge with no significant overfitting, extending training is unlikely to improve results. Instead, tuning hyperparameters is a more promising path.

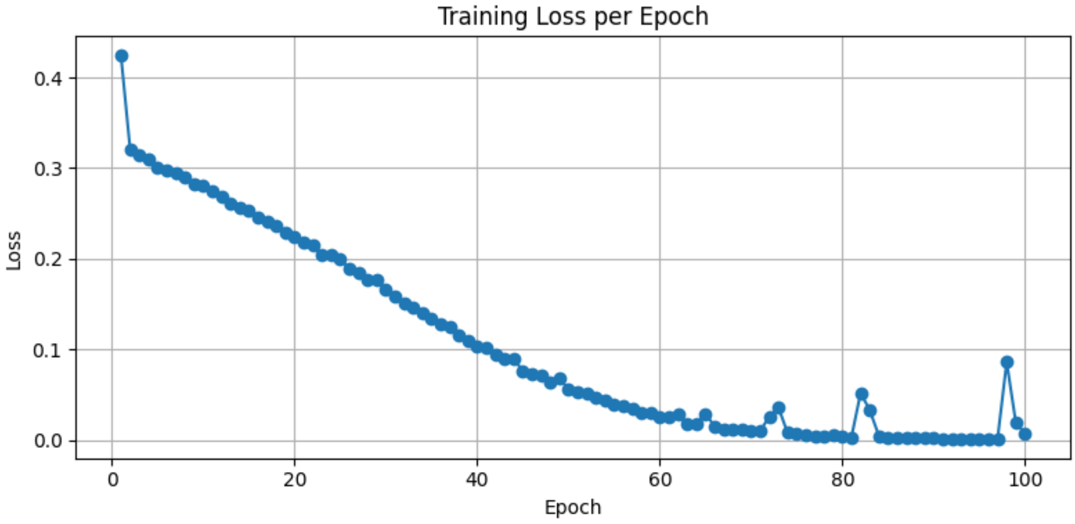


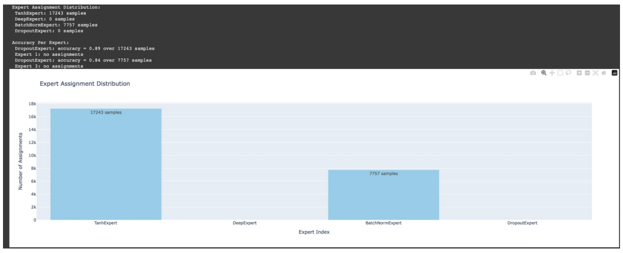
Figure 16: Training loss per epoch over 100 epochs. The curve flattens near the end with occasional spikes.

7.3 Analyzing Expert Assignment Behavior

To better understand how the gating network distributes inputs across experts, we examined the assignment distribution for the model *DistilBERT + 4 Experts (Tanh, Deep, BatchNorm, Dropout) + Noisy Gating*, trained on the full dataset. As shown in Figure **??**, the gating network exhibits a clear bias toward certain experts—most notably, the `TanhExpert`, which receives a significantly higher number of assignments compared to the others. Meanwhile, some experts, such as `DeepExpert` and `DropoutExpert`, receive no assignments at all.

This imbalance suggests that with $K = 2$ (top-2 sparse expert selection), the model is not exploring the available expert space effectively. The gating network fails to learn a diverse or balanced routing policy, potentially undermining the core idea of a Mixture of Experts. This behavior could be attributed to insufficient training of the router or to an overly confident gating mechanism that settles early on specific experts.

These findings highlight the importance of further investigating the effect of K, regularization of the gating output, and training dynamics of the router, to ensure that all experts are utilized and the model benefits from the intended modular structure.



```
Expert Assignment Distribution:
 TanhExpert: 17243 samples
 DeepExpert: 0 samples
 BatchNormExpert: 7757 samples
 DropoutExpert: 0 samples

Accuracy Per Expert:
 DropoutExpert: accuracy = 0.89 over 17243 samples
 Expert 1: no assignments
 DropoutExpert: accuracy = 0.84 over 7757 samples
 Expert 3: no assignments
```

Figure 17: Expert assignment distribution showing dominance of `TanhExpert` in routing decisions.

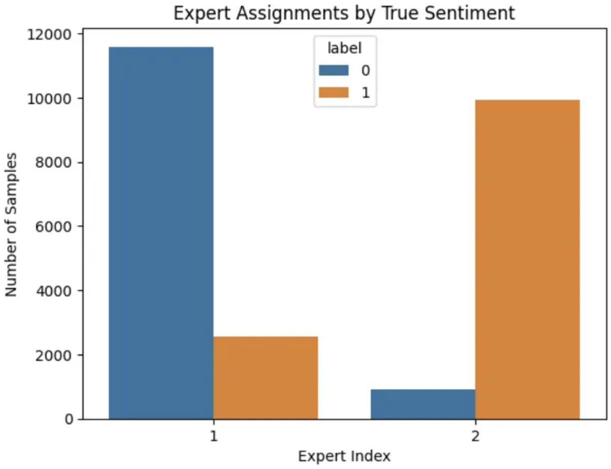


Figure 18: Expert assignments by true sentiment labels showing varying preferences.

8 Conclusion

Ce projet a permis d'explorer en profondeur les différentes étapes d'une tâche de classification de sentiments appliquée aux critiques de films IMDb, allant de la préparation des données jusqu'à l'implémentation avancée d'un modèle de type Mixture of Experts (MoE). Grâce à l'intégration d'embeddings issus de BERT et à la combinaison d'experts spécialisés (MLP, CNN, LSTM), nous avons obtenu des performances compétitives, avec une précision atteignant jusqu'à 88.5%. Ces résultats confirment l'intérêt des approches hybrides et modulaires pour capturer la complexité du langage naturel. En dépit de quelques limitations, notamment liées au déséquilibre dans l'activation des experts, les expérimentations menées ouvrent la voie à de futures améliorations fondées sur le réglage fin des mécanismes de routage et l'optimisation de l'architecture globale.

9 Perspectives and Future Work

To overcome these limitations, several directions can be explored:

- **Data Augmentation and Transfer Learning:** Leveraging techniques such as back-translation, synonym replacement, or pretraining on related tasks can mitigate data scarcity.
- **Fine-Tuned Transformer Models:** Rather than using generic BERT embeddings, fine-tuning transformer models on the specific task could significantly boost performance.
- **Dynamic Expert Routing:** Incorporating reinforcement learning or attention-based gating mechanisms may allow for more intelligent and adaptive expert selection in the MoE.
- **Multilingual and Cross-Domain Extension:** Extending the approach to multilingual sentiment analysis or domain adaptation (e.g., reviews from different sectors) would broaden its applicability.
- **Human-in-the-Loop Approaches:** Integrating human feedback for model correction or semi-supervised learning could enhance both performance and trustworthiness.