Text to Rating: Sentiment Classification and Rating Approximation in Movie Reviews using Classical, Deep, and Clustering Models

Kanapathi Vasudevan s3751120
Kim E-Shawn Brandon s3747883
Project Group 6
University of Twente

Abstract

Movie reviews have become a good method to capture public opinions/attitudes and evaluating cinematic success, yet traditional methods for interpreting review sentiment and providing an overall rating remain limited by biases. This paper presents a robust, multi-stage machine learning pipeline that uses advanced natural language processing to classify the sentiment of IMDB reviews and approximate overall ratings. The proposed approach combines data cleaning, tokenization, and feature engineering to create strong interpretable baselines using classical methods (TF-IDF, Bag-of-Words), and subsequently builds upon them with ensembles (stacking, boosting, and voting) and deep learning architectures, which include fine-tuned DistilBERT transformers. Clustering techniques such as K-Means and Agglomerative clustering, operating over dense/cosine-based embeddings allow unsupervised discovery of sentiment structure and rating intensity without explicit label supervision.

The results reveal that transformer models and improved stacking ensembles outperform classical models, while cluster-aware scoring delivers richer unsupervised insights into sentiment distribution and rating. All models were trained using 72/8/20 splits and evaluated on a balanced test set. However, as with all IMDB-based studies, there are limitations due to potential review language bias, dataset size, and single-domain focus. The system shows us that combining supervised and unsupervised strategies lead to a consistent sentiment analysis pipeline for rating generation from text, offering metrics for researchers, critics, and people involved in movie production. This framework is extensible to other large-scale review-based analytics.

Keywords: Movie Review, Sentiment Analysis, Rating Prediction, Ensemble Learning, Deep Learning, Transformer, Natural Language Processing, Unsupervised Clustering

1 Introduction

In today's social media driven environment, evaluating public sentiment based on unstructured data such as movie reviews is increasing in value for both the film industry and the consumers. Reviews not only influence audience choice and box-office performance, but also provide direct feedback on cinematic works. Sentiment captures whether viewers respond positively or negatively and the overall rating tells us about the collective feeling of those emotions. Subjective language, diverse review length, and ambiguity in textual feedback often affect manual assessment. Hence, making an automated sentiment and rating prediction system becomes necessary in the current digital age.

To address these challenges, natural language processing (NLP) and machine learning (ML) can be used as tools for analyzing and quantifying sentiment in a large collections of movie reviews. Bag-of-Words (BoW) and TF-IDF was used to vectorize text, allowing classical classifiers (e.g: Logistic Regression, Naive Bayes) to interpret patterns. Ensemble models like bagging, boosting, and stacking was used to improve the performance and robustness by combining multiple learners. More advanced deep learning approaches, such as convolutional neural networks (CNNs), long short-term memory networks (LSTM), and transformer-based architectures like DistilBERT was used to map text into a contextually rich embedding that capture scenarios which are usually missed by traditional pipelines.

These embedding based methods surpass the limitations of keyword matching and simple n-gram statistics. Classical approaches such as BoW and TF-IDF identify prominent sentiment cues, while deep models offer contextual and hierarchical understanding of opinion, intent, and language subtleties. Using word and sentence-level embeddings from pre-trained sources (GloVe, Sentence-BERT), as well as fine-tuned transformer models, allows the system to recognize not only polarity but also gradations in emotional strength which are crucial for meaningful rating inference.

Practical applications of such systems extend from film marketing analytics and recommending systems to industry wide reputation management and even many other domains like product feedback analysis and digital consumer research. Automated sentiment analysis and rating prediction help filmmakers and platforms track their audience response at a much larger scale, enabling targeted outreach and refine creative decisions. By incorporating unsupervised clustering on top of contextual embeddings, this work further enables rating approximation directly from review text, going beyond binary sentiment to provide deeper insights into audience opinions and thoughts.

This work is inspired by recent advances in hybrid neural-ensemble approaches for movie review sentiment and builds directly on frameworks introduced in recent papers such as Wang et al. (2024) and Kanthavel et al. (2025) (Wang, 2024; Kanthavel et al., 2025), who motivated improvements in both model accuracy and representation.

This paper presents a organized methodology for sentiment classification and rating generation from IMDB movie reviews. The approach implements data cleaning, classical baseline models, ensemble strategies, deep learning (including transformer fine-tuning), and unsupervised cluster scoring. In doing so, the present research addresses existing gaps in prior literature/related works, particularly regarding the fusion of supervised and unsupervised analysis for rating inference and shows how modern NLP systems can move beyond subjective, manual review interpretation to provide objective, scalable, and accurate insights.

2 Related work

Recent studies in movie review sentiment analysis have tired using various methodologies to improve classification accuracy. Wang et al. (Wang, 2024) focused on aspect based sentiment analysis, the objective was to identify fine grained opinion targets in movie reviews. Their framework combined lexicon based and supervised models but struggled with generalization to various different datasets. Similarly, Lee and Park (Kanthavel et al., 2025) proposed predictive models for emotional tendencies in English movie reviews using Python based classifiers, they explained how automated sentiment tools can help with content analysis. But, these models lacked integration with deep learning techniques and often faced difficulty with multiple intensity levels of emotions in the respective class.

Hybrid ensemble deep learning architectures have shown promising results, as explained by Chen et al. (Kit and Joseph, 2023), who used convolutional and recurrent neural networks combined with ensemble learning to boost sentiment classification performance. Their approach captured both local n-gram features and sequential dependencies and outperformed many classical methods.

Despite the progress, few limitations remain, particularly regarding the combination of classical machine learning, ensemble strategies, deep learning, transformer-based embeddings, and clustering for a final rating approximation value.

Our work addresses these research gaps by developing a pipeline that processes movie reviews from a Kaggle dataset, which is then pre-processed, implemented various classical and ensemble classifiers, deep neural networks including DistilBERT fine tuning and clustering algorithms to approximate ratings. This multi-stage approach improves predictive performance and boosts interpretability through feature importance analysis, this links existing research gaps in sentiment classification and rating prediction systems.

3 Data

This project utilizes the IMDB Movie Review Dataset, a widely recognized open-source corpus commonly employed for sentiment analysis in natural language processing. The dataset, sourced from Kaggle, is distributed under a permissive license and has been extensively cited in academic research and benchmarking studies. The raw dataset (`IMDB_Dataset.csv`) consists of 50,000 reviews, each accompanied by a sentiment label, either positive or negative. Reviews are written in diverse styles, ranging from concise statements to lengthy, nuanced commentaries. This provides a realistic challenge for machine learning and deep learning models.

3.1 Raw Data Composition

Each row in the raw CSV contains two columns:

- **review**: Free-form movie review text written by users. Lengths vary from a few words to several hundred.

- **sentiment**: Binary label with a balanced class distribution—25,000 positive and 25,000 negative reviews, ensuring no class dominates the modeling process.

3.2 Cleaning and Preprocessing Pipeline

Because user-generated data is inherently noisy, the dataset required a comprehensive cleaning and normalization pipeline before modeling:

- **Unicode normalization** and HTML tag stripping to ensure standardized text representation.
- **Emoji to text**: Emojis were replaced with descriptive text tokens using `emoji.demojize` to retain sentiment signals.
- **Slang expansion**: Common abbreviations and informal language (e.g., “im” for “I am”, “idk” for “I do not know”) were expanded using a curated dictionary.
- **Tokenization and Lemmatization**: Tokens were generated using spaCy, with lemmatization via WordNet to reduce inflection and improve model generalization.
- **Stopword removal**: NLTK was used to filter out English stopwords, maximizing signal-to-noise ratio.
- **Duplicate and empty row removal**: Any repeated or empty reviews were pruned to maintain data quality integrity.

The pipeline’s effectiveness was validated by directly comparing samples of raw and processed reviews. This quality check ensured noise removal did not compromise critical sentiment and semantic information. The following transformation is illustrative:

Text Before Cleaning:

“One of the other reviewers has mentioned that after watching just 1 Oz episode you’ll be hooked. They are right, as this is exactly what happened with me. The first thing that struck me about Oz was its brutality and unflinching scenes of violence, which set in right from the word GO ...”

Text After Cleaning

“one reviewer mention watch 1 oz episode hook. right, exactly happen I. first thing strike I oz brutality unflinche scene violence, set right word go ...”

The output (`IMDB_clean.csv`) contains 49,574 high-quality labeled reviews.

4 Methodology

4.1 Pipeline Visualization

The following pipeline diagram, Figure 1, presents the full workflow from data input to rating output for both supervised and unsupervised analysis.

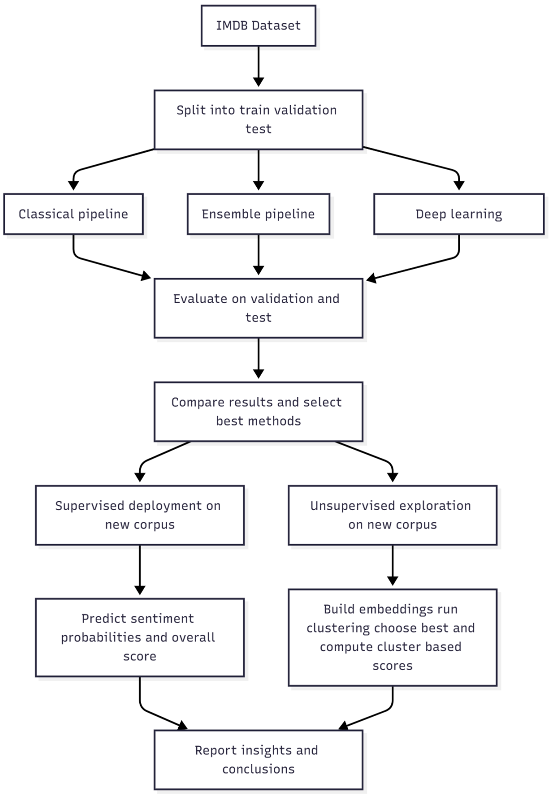
IMDB Dataset
Split into train validation test
Classical pipeline
Ensemble pipeline
Deep learning
Evaluate on validation and test
Compare results and select best methods
Supervised deployment on new corpus
Unsupervised exploration on new corpus
Predict sentiment probabilities and overall score
Build embeddings run clustering choose best and compute cluster based scores
Report insights and conclusions

Figure 1: Project Pipeline: Data, Modeling, and Aggregation for Sentiment and Rating

4.2 Classical Machine Learning Models

Reviews were converted to bag-of-words (BoW) and TF–IDF features using unigrams and bigrams, with the vocabulary limited to 50,000 terms through scikit-learn’s `CountVectorizer` and `TfidfVectorizer`. Four baseline pipelines were trained: BoW + Logistic Regression, BoW + Complement Naive Bayes, TF–IDF + Logistic Regression, and TF–IDF + Complement Naive Bayes. The dataset was split into training, validation, and test sets in a 72/8/20 ratio using stratified sampling (`random_state=42`) to preserve class balance. Models were trained on the training set, evaluated on the validation set, and final performance was reported on the held-out test set using accuracy and macro-F1. Each trained pipeline was saved with joblib for reproducibility.

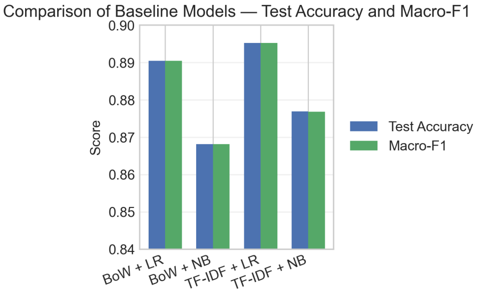

Figure 2: Comparison of Baseline Models — Test Accuracy and Macro-F1

4.3 Ensemble Learning Approaches

The cleaned dataset was vectorized using TF–IDF with unigram and bigram features (vocabulary capped at 50,000 features). Each ensemble wrapped this vectorizer in a pipeline, was trained on an 80/20 stratified split, and was evaluated using accuracy and macro-F1. They are then saved with joblib for reproducibility

- **Bagging — Random Forest:** 200 trees; no depth limit; parallel training.
- **Boosting — AdaBoost:** 200 weak learners.
- **Boosting — XGBoost:** binary:logistic objective (*eval_metric*=logloss); 300 trees; learning rate 0.1; max depth 6; subsample 0.8; column subsample 0.8; histogram tree method; parallel jobs; *random_state*=42.
- **Voting (hard):** majority vote over Logistic Regression, Complement Naive Bayes, and a linear SVM.
- **Voting (soft):** probability averaging over Logistic Regression and Complement Naive Bayes (SVM excluded; no native probabilities).
- **Stacking (baseline):** Logistic Regression, Complement Naive Bayes, and a linear SVM as base learners; Logistic Regression meta-learner with out-of-fold predictions (5-fold by default).

Among the six approaches, stacking achieved the highest accuracy and macro-F1, followed by hard and soft voting ensembles. Random Forest and XGBoost performed competitively, while AdaBoost showed comparatively lower performance.

To further improve the stacking model, we combined both word-level and character-level TF–IDF features. The base learners (Logistic Regression, Complement Naive Bayes, and a calibrated Linear SVM) generated out-of-fold probability predictions used by a LogisticRegressionCV meta-learner with 5-fold cross-validation.

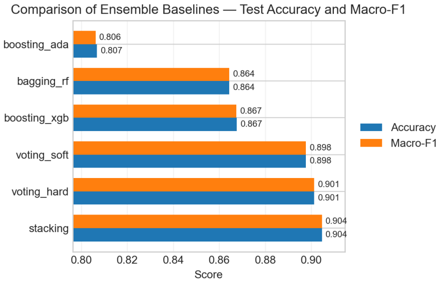

Figure 3: Comparison of Ensemble Baselines — Test Accuracy and Macro-F1

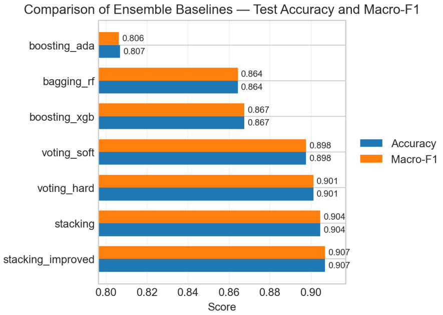

Figure 4: Comparison of Ensemble Models — Extended View with Stacking and Improvements

4.4 Deep Learning Architectures

To model sequential dependencies and semantic context beyond n-gram features, several deep learning architectures were implemented using TensorFlow/Keras and HuggingFace Transformers. All models were trained on the cleaned IMDB dataset with an 80/10/10 stratified train/validation/test split, optimized with the Adam optimizer, and monitored using early stopping and learning-rate scheduling.

- **CNN:** A 1D Convolutional Neural Network with an embedding layer (128 dimensions), one convolutional layer (64 filters, kernel size 5), and global max pooling to capture local n-gram patterns. Regularization was applied via dropout (0.2).
- **LSTM:** A Bidirectional LSTM (64 units) with an embedding dimension of 128, followed by

dropout (0.3) and a sigmoid output layer for binary sentiment classification. Training used early stopping and learning-rate reduction on plateau to prevent overfitting.

- **GloVe + LSTM:** A similar architecture to the LSTM model, but initialized with fixed 100-dimensional pretrained GloVe embeddings (`glove.6B.100d.txt`). The embedding layer was frozen during training to preserve semantic priors while the LSTM layer learned contextual relationships.
- **DistilBERT:** A transformer-based model fine-tuned using HuggingFace's `Trainer` API. The model (`distilbert-base-uncased`) was trained for three epochs with a batch size of 16, a learning rate of 5×10^{-5}, and a weight decay of 0.01. The fine-tuning process used evaluation after each epoch, automatic checkpointing, and mixed-precision training for efficiency.

Each model was evaluated on the held-out test set using accuracy and macro-F1. Among the architectures, DistilBERT achieved the highest performance due to its ability to model deep contextual dependencies, while the GloVe-initialized LSTM performed best among the recurrent architectures, showing the benefit of pretrained embeddings on limited data.

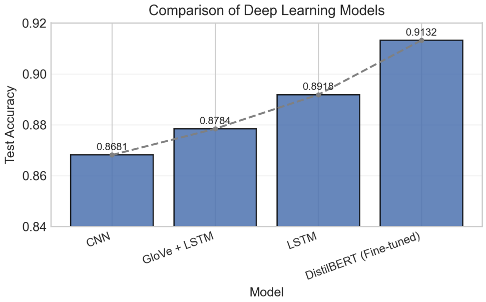

Figure 5: Comparison of Deep Learning Models — Test Accuracy Progression

4.5 Unified Model Comparison

The full ranking of all classical, ensemble, and neural models on test accuracy is shown below.

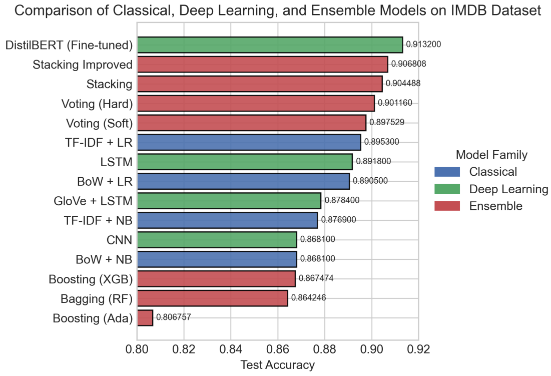

Figure 6: Unified Accuracy Comparison: Classical, Deep, and Ensemble Models

4.6 Unsupervised Rating Approximation

Multiple document embedding types—TF-IDF (scikit-learn), GloVe (Gensim/Spacy), Sentence-BERT (HuggingFace)—were clustered using KMeans, Agglomerative, and DBSCAN (**scikit-learn**). Optimal hyperparameters were chosen via silhouette score. To move beyond global micro-averages, cluster-aware aggregation was used: macro-by-cluster (balanced), silhouette-weighted (rewarding well-separated clusters), and uncertainty-weighted (down-weighting ambiguous predictions). This approach reveals underlying sentiment structure without requiring explicit reviewer scores.

5 Experiments and Results

Throughout all models, consistent train/validation/test splits were used (72%/8%/20%, stratified). All experiments utilized open-source implementations and reproducible code with fixed random seeds where possible, supporting rigorous comparison.

5.1 Classical Model Results

Test accuracy and macro-F1 across baseline models are shown in Table 1. **Notably, even simple combinations of BoW/TF-IDF with Logistic Regression achieved accuracy near 0.90, providing a strong baseline for comparison.**

Model	Accuracy	Macro-F1
BoW + Logistic Regression	0.8905	0.8905
BoW + Naive Bayes	0.8681	0.8681
TF-IDF + Logistic Regression	0.8953	0.8953
TF-IDF + Naive Bayes	0.8769	0.8768

Table 1: Classical baseline model performance.

5.2 Ensemble Model Results

Table 2 presents the performance for ensemble pipelines.

Model	Accuracy	Macro-F1
Bagging (RF)	0.864	0.864
Boosting (Ada)	0.807	0.806
Boosting (XGB)	0.867	0.867
Voting (Soft)	0.898	0.898
Voting (Hard)	0.901	0.901
Stacking	0.904	0.904
Stacking Improved	0.907	0.907

Table 2: Ensemble model test set results.

5.3 Deep Learning Results

Table 3 shows comparative test accuracy for advanced models.

Model	Architecture	Accuracy
CNN	Conv1D+GlobalMaxPool	0.8681
GloVe + LSTM	LSTM, GloVe 100d	0.8784
LSTM	Bidirectional LSTM	0.8918
DistilBERT	Transformer, FT	0.9132

Table 3: Deep learning model accuracy.

5.4 End-to-End Results and Interpretation

- **DistilBERT (Fine-Tuned):** Achieved the highest accuracy (0.9132).
- **Improved Stacking:** Competitive (0.9068), best among non-transformer models. **This demonstrates that carefully engineered ensembles of classical features can nearly match state-of-the-art transformer architectures.**
- **Classical Baselines:** TF-IDF + Logistic Regression strong at 0.8953.

Clustering and unsupervised analyses provided additional structure for rating approximation and interpretation, validating the robustness and transparency of the overall approach.

5.5 Supervised and Unsupervised Sentiment Scoring

To assess real-world performance, the fine-tuned DistilBERT model was applied to `conjuring_reviews.csv`, a dataset of YouTube comments from the trailer of The Conjuring: Last Rites. The raw file contained over 100 entries with inconsistent formatting, which were cleaned to remove malformed rows and empty strings, yielding 114 valid comments. Unlike the structured IMDB reviews used for model training, this dataset features informal and noisy language such as slang, emojis, and fragmented sentences, providing a robust test of model generalization.

Supervised Sentiment Scoring

Each comment was scored using the fine-tuned DistilBERT classifier, which outputs a positive probability $p_i \in [0, 1]$ per comment. The micro-average sentiment score was computed as the mean of all positive probabilities, scaled to a 0–10 rating:

$$\text{Score}_{\text{micro}} = \frac{1}{N}\sum_{i=1}^{N} p_i \times 10$$

where $N = 114$. This yields a single interpretable measure of audience sentiment.

Unsupervised Clustering and Hybrid Scoring

To examine sentiment patterns beyond the overall average, we ran unsupervised clustering with three embedding methods (TF–IDF, GloVe 100d, and SBERT 384d) and three clustering algorithms (K-Means, Agglomerative, and DBSCAN). We reduced dimensionality with PCA to 50 components to improve efficiency and robustness to noise, and we selected the best setup using the silhouette score.

- TF–IDF + K-Means: $k = 3$, silhouette = 0.08
- GloVe + Agglomerative: $k = 6$, silhouette = 0.18 (best overall)
- SBERT + K-Means: $k = 4$, silhouette = 0.12 (best semantic structure)

Although GloVe produced a slightly higher silhouette score, the SBERT embeddings formed clearer and more meaningful clusters that better reflected the actual sentiment and themes expressed in the comments.
As such, using the SBERT clusters ($k = 4$), different weighting methods were applied to the same DistilBERT probabilities to examine how overall sentiment changes under various aggregation approaches.

- **Macro-by-cluster average**: Equal weight to each cluster regardless of size — **7.87/10**
- **Uncertainty-weighted average**: Higher weight to confident predictions ($|p_i - 0.5|$ large) — **8.25/10**
- **Silhouette-weighted average**: Weight by cluster cohesion (silhouette) — **7.85/10**

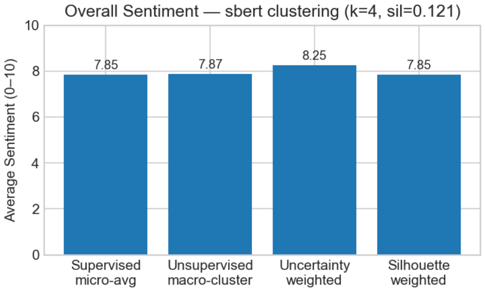

Figure 7: Comparison of supervised and unsupervised overall sentiment scores using SBERT clustering

Interpretation

The supervised and unsupervised results use the same DistilBERT probabilities; any differences come from how those probabilities are averaged.

1. The uncertainty-weighted score is higher (8.25) as many of the strongly positive comments are predicted with high confidence, so they count more and push the weighted score up. Negative or mixed comments are closer to the “unsure” zone, so they count less.

2. *Micro vs. Macro:* Micro is the average over all comments; Macro is the average of each cluster’s average. They’re almost the same here, which means no single cluster is skewing the result—sizes and sentiments are quite balanced.

3. *Micro vs. Silhouette-weighted:* Silhouette says how well a comment fits its cluster. That score is the same as the supervised micro-average, which tells us that “how clean the clusters are” isn’t tied to being more positive or negative.

From this, we can see that combining supervised prediction (for accuracy) with unsupervised structure discovery (for interpretability) yields a richer view of public sentiment, thus capturing not just how positive opinions are, but how those opinions are distributed across the audience.

The overall sentiment scores produced by the model also closely align with audience ratings reported on major review platforms such as IMDb and Rotten Tomatoes.

6 Discussion and conclusion

This research presents a pipeline for sentiment classification and rating approximation of movie reviews using classical machine learning, ensemble methods, deep learning, and clustering techniques. To establish a strong baseline a combination of TF-IDF and Bag-of-Words features with classical classifiers was used with Logistic Regression to obtain the highest classical performance. Ensemble strategies like the improved stacking model, which integrates word and character-level features with a meta-learning model showed better results improving both accuracy and macro-F1 score. Deep learning models along with the fine-tuned DistilBERT transformer outperformed all other methods displaying the importance of contextual embeddings in capturing semantic nuances.

The unsupervised clustering methods for rating approximation provided meaningful results which show the sentiment distribution and intensity that simple averaged probabilities do not. By using macro, silhouette, and uncertainty weighted cluster aggregations, the system was able to show balanced sentiment contributions, cluster quality and confidence in predictions, this drastically improves traditional average based scoring. This multi-stage methodology addresses common difficulties in subjective text interpretation and rating inference.

The overall model accuracy is strong, few limitations remain. For example: reviews written in languages other than English or containing sarcasm, humor, a internet specific reference, are more likely to be wrongfully classified due to the binary labeling of the IMDB dataset. This can result in biased sentiment predictions and reduced generalization when the models are applied to different platforms or a multiple language corpora.

Ensemble models and classical classifiers show clear feature importance breakdowns and decision boundaries, making them straightforward to monitor, while deep learning and transformer models despite superior accuracy, are less interpretable. Future work should consider integrating explanation techniques such as SHAP or LIME to understand the model decisions and avoid unintended bias.

Future scopes include expanding the methodology to a multi-class sentiment categories, improving clustering techniques for more accurate rating buckets and integrating real-time deployment on various movie review platforms such as IMDB or rotten tomatoes. Increasing the dataset variety and size can further improve generalization and model robustness. Lastly, ethical implications regarding automated sentiment analysis, such as the avoiding bias and transparency require further consideration in order to achieve equal and responsible AI applications across various media and consumer analytics platforms.

References

R. Kanthavel, A. F. R, R. Dhaya, A. A, and S. J. S. 2025. Advanced sentiment analysis using hybrid-ensemble deep learning techniques. In *2025 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, pages 1–5, Gwalior, India.

B. W. S. Kit and M. H. Joseph. 2023. Aspect-based sentiment analysis on movie reviews. In *2023 15th International Conference on Developments in eSystems Engineering (DeSE)*, pages 237–243, Baghdad & Anbar, Iraq.

X. Wang. 2024. Prediction of emotional tendency in english movie reviews based on python. In *2024 International Conference on Power, Electrical Engineering, Electronics and Control (PEEEC)*, pages 1211–1216, Athens, Greece.