IMDB Movie Genre Classification: Final Report

Name:Shreyas Srinivas Bikumalla

Abstract

This project explores the complex task of movie genre classification using IMDb titles and descriptions. Given the multi-class nature of the problem and challenges like overlapping genre semantics, class imbalance, and noisy user-generated text, we design a robust classification pipeline. The dataset spans **27 genres**, with highly skewed distributions genres like Drama and Documentary dominate, while others like Musical and Game-Show are underrepresented. To manage the high dimensionality of textual features, we apply TF-IDF vectorization and reduce dimensionality using PCA and LDA.

We evaluate a wide range of models, including traditional classifiers (Logistic Regression, Naive Bayes, SVM, Decision Trees, Random Forests, XGBoost), a custom Bayesian classifier, and deep learning models (CNN, LSTM). Performance is assessed through accuracy, cross-validation variance, and detailed confusion matrices. Visualization through PCA and LDA plots helps interpret genre separability.

1. Introduction

The ability to automatically determine a movie's genre from its metadata holds significant value in domains such as recommendation systems, streaming content tagging, and information retrieval. Genre classification involves understanding natural language inputs (titles and descriptions) and mapping them to predefined labels. Despite its seemingly simple framing, this task is complex due to the multi-class nature, subjective genre boundaries, and heterogeneous language used in movie synopsis.

In this project, we explore **single label genre prediction** based on IMDb titles and plot descriptions. While movies can belong to multiple genres in reality, our classification framework simplifies the task by predicting the most probable genre per movie. The dataset consists of 27 distinct genre classes, making this a large-class, imbalanced classification problem.

Motivation

Movie descriptions can be highly variable, some are poetic or sarcastic, others are sparse or overly detailed. Moreover, genres frequently co-occur, making the classification boundary fuzzy. Traditional models such as Logistic Regression may struggle to generalize across such semantic

ambiguity, while **deep learning models**, though powerful, risk overfitting, especially when certain classes are underrepresented.

To navigate this complexity, we propose a pipeline that:

- Cleans and vectorized text using TF-IDF.
- Applies dimensionality reduction (PCA , LDA).Trains and compares multiple classifiers across raw, PCA-transformed, and LDA-transformed data.
- Quantifies model performance using both accuracy and 5-fold cross-validation error variance.
- Visualizes class separability via PCA and LDA projections.

2. Dataset Description:

- **Source:** Kaggle "Genre Classification Dataset – IMDB"
- **Train Data:** 35,000 labeled rows with fields - ID, TITLE, DESCRIPTION, GENRE
- **Test Data:** 7,000 rows without GENRE
- **Solution Data:** Answer for test data used for final evaluation
- **Validation Split:** 10% (3,500 samples) of training set used to tune model hyperparameters
- **Format:**

```
Train data:
ID ::: TITLE ::: GENRE ::: DESCRIPTION
ID ::: TITLE ::: GENRE ::: DESCRIPTION
ID ::: TITLE ::: GENRE ::: DESCRIPTION
ID ::: TITLE ::: GENRE ::: DESCRIPTION

Test data:
ID ::: TITLE ::: DESCRIPTION
ID ::: TITLE ::: DESCRIPTION
ID ::: TITLE ::: DESCRIPTION
ID ::: TITLE ::: DESCRIPTION
```

Features and Labels:

- Title and Description were used as text features
- Genre is the target label, encoded using **LABELENCODER()**

Genres (27 Total):

1)Adventure 2) Animation 3) Biography 4) Comedy 5) Crime 6) Documentary 7) Drama 8) Family 9) Fantasy 10) History 11) Horror 12) Music 13) Musical 14) Mystery 15) News 16) Reality-TV 17) Romance 18) Sci-Fi 19) Short 20) Sport 21) Talk-Show 22) Thriller 23) War 24) Western 25) Action 26) Adult 27)Game-Show

Models Experimented (9 in total):

Traditional models : Logistic Regression , Multinomial Naive Bayes , Support Vector Machine (SVM) , Decision Tree , Random Forest , XGBoost , Custom Bayesian Classifier

Deep learning models: Convolutional Neural Network (CNN) , Long Short-Term Memory Network (LSTM)

3. Pre-processing and Feature Engineering

The preprocessing phase is critical in transforming the raw, unstructured IMDb movie descriptions into a format suitable for machine learning algorithms. Our goal was to clean, normalize, and vectorize the data in a way that preserved semantic information while reducing noise and redundancy. Below is a detailed breakdown of each preprocessing component and the rationale behind it.

3.1 Text Cleaning

Movie descriptions in the raw dataset often contained extraneous elements such as HTML tags , URLs, and various punctuation symbols. These artifacts contribute noise without adding semantic value. To remove them, we applied regular expressions REGEX module to:

- Strip out HTML tags
- Eliminate non-alphanumeric characters
- Remove extra whitespaces and digits

This cleaning step ensured the resulting text was concise and consistent across entries.

3.2 Lowercasing

All words were converted to lowercase to ensure uniformity. For instance, "Love" and "love" would otherwise be treated as two separate tokens by the tokenizer. Lowercasing helps reduce vocabulary size and improves generalization.

3.3 Stopword Removal

We removed common English stopwords ("the", "is", "in", "of") using the nltk.corpus.stopwords list. These words appear frequently but carry minimal discriminative power for classification. Removing them helps focus the model on more meaningful terms.

3.4 Lemmatization

To reduce words to their base forms ("running" to "run", "films" to "film"), we applied WordNetLemmatizer from NLTK. Unlike stemming, lemmatization preserves actual dictionary words and thus maintains semantic clarity while consolidating similar forms.

3.5 Tokenization & Vectorization using TF-IDF

We employed TfidfVectorizer from scikit-learn to convert cleaned textual data into numeric feature vectors. This method weighs words based on their importance in a document relative to the corpus. Specifically:

- **Full Feature Model**: Used 5000 most frequent terms for standard training and raw classifier comparisons.
- **Reduced Feature Model**: Limited to 2000 features for experiments involving PCA and LDA to reduce computational load and increase numerical stability.

This dual TF-IDF approach allowed us to evaluate model performance on both high-dimensional and compressed representations.

3.6 Feature Scaling for PCA/LDA

Before applying dimensionality reduction techniques like PCA and LDA, we standardized the TF-IDF vectors using StandardScaler. Since PCA and LDA are sensitive to feature magnitude, normalization ensures that all features contribute equally during transformation. Scaling was only applied to the 2000 feature matrix, as PCA/LDA was not performed on the 5000-feature vectors.

3.7 Dual Feature Set Strategy

We explicitly maintained two feature sets throughout our experimentation:

- A **5000-dimensional TF-IDF matrix** used in raw TF-IDF models (Logistic Regression, SVM, XGBoost, Decision Tree, Random Forest, Naive Bayes)

- A **scaled 2000-dimensional TF-IDF matrix** which was transformed using:

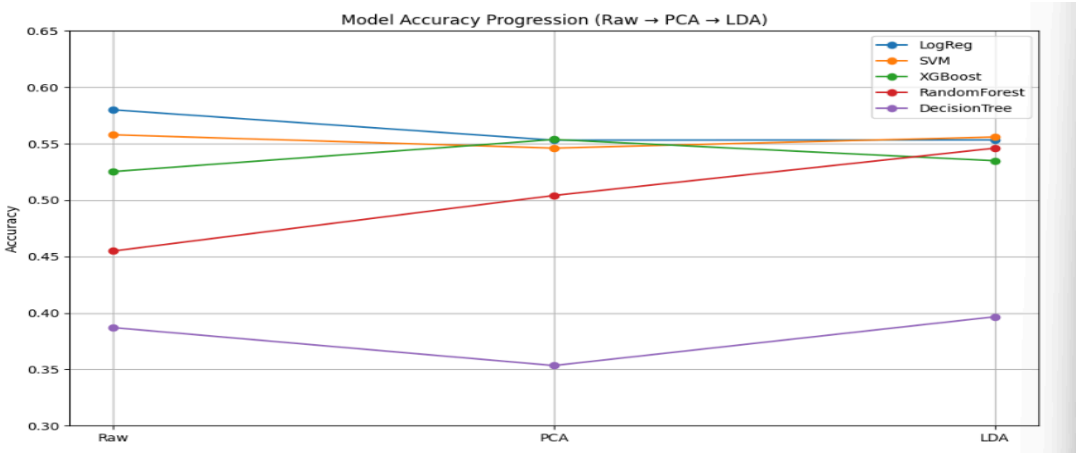
- **PCA (Principal Component Analysis)** – 100 components
- **LDA (Linear Discriminant Analysis)** – up to c-1 components.

3.8 Data Normalization:

In our pipeline, normalization was applied to the lower-dimensional TF-IDF matrix used for PCA and LDA experiments. Since both PCA and LDA are sensitive to the scale of input features, we used StandardScaler from sklearn.preprocessing to standardize the 2000-dimensional TF-IDF vectors before reduction. This ensured that each feature contributed equally to the dimensionality reduction process and prevented bias toward high-frequency terms.

- StandardScaler() was applied to the 2000-feature TF-IDF vectors (X_small and X_small_test) before PCA/LDA.
- Normalized data helped PCA preserve meaningful global variance and allowed LDA to project samples into a discriminative 26D space.
- Without normalization, PCA and LDA were unstable and sometimes led to poor generalization or distorted cluster formation in the projection plots.

4. Classifier Breakdown and Analysis: All classifiers were evaluated on accuracy, confusion matrices, and cross-validation variance.Here below is just an overview of progression and details explained later.



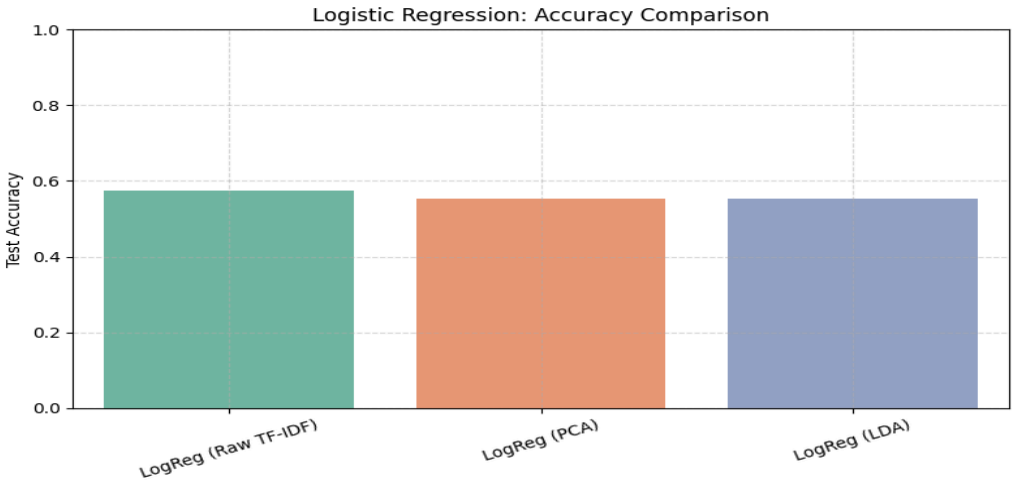
4.1 Logistic Regression:

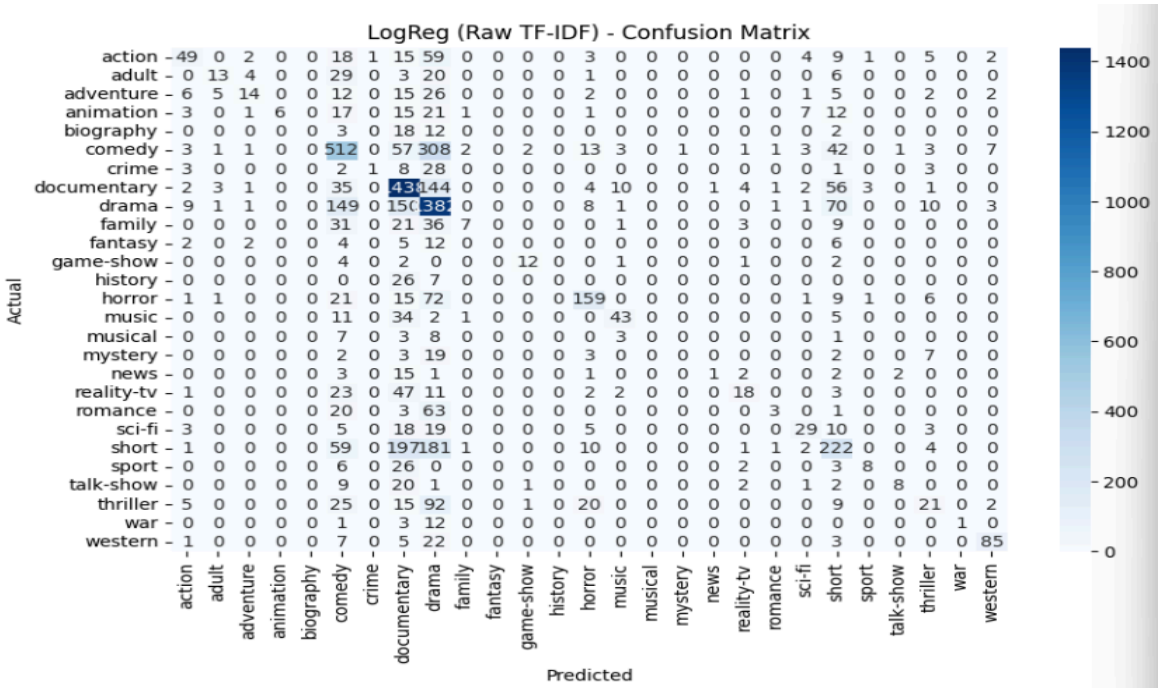
Logistic Regression is a widely used linear classifier that models the log odds of each class based on weighted input features. It is particularly effective when dealing with high-dimensional,

sparse datasets such as those produced by TF-IDF vectorization. In our project, logistic regression performed well using raw TF-IDF features, achieving an accuracy of approximately 57.6% on the test set. It showed consistent results with a cross-validation accuracy of 56.39%, indicating low variance and good generalization.

When PCA was applied to reduce dimensionality, accuracy decreased slightly to 55.31%, suggesting some loss of discriminative power due to unsupervised compression. In contrast, the LDA-based variant achieved a very similar test accuracy of 55.33%, but significantly higher cross-validation accuracy at 64.19%. This indicates that LDA, by projecting data into class-separating directions, may better capture structure within the training data, albeit at the risk of overfitting certain class boundaries.

Across all variants, logistic regression handled dominant genres such as Drama, Documentary, and Comedy reasonably well, likely due to their strong representation in the dataset. However, its performance on rare classes such as Musical, Biography, and War was poor, with many of these instances either misclassified into more frequent genres or completely ignored. The confusion matrices reveal frequent misclassification between semantically similar genres, such as Biography and Drama, or Thriller and Mystery. Overall, logistic regression provides a reliable baseline for multi-class genre classification, particularly in combination with raw TF-IDF features.



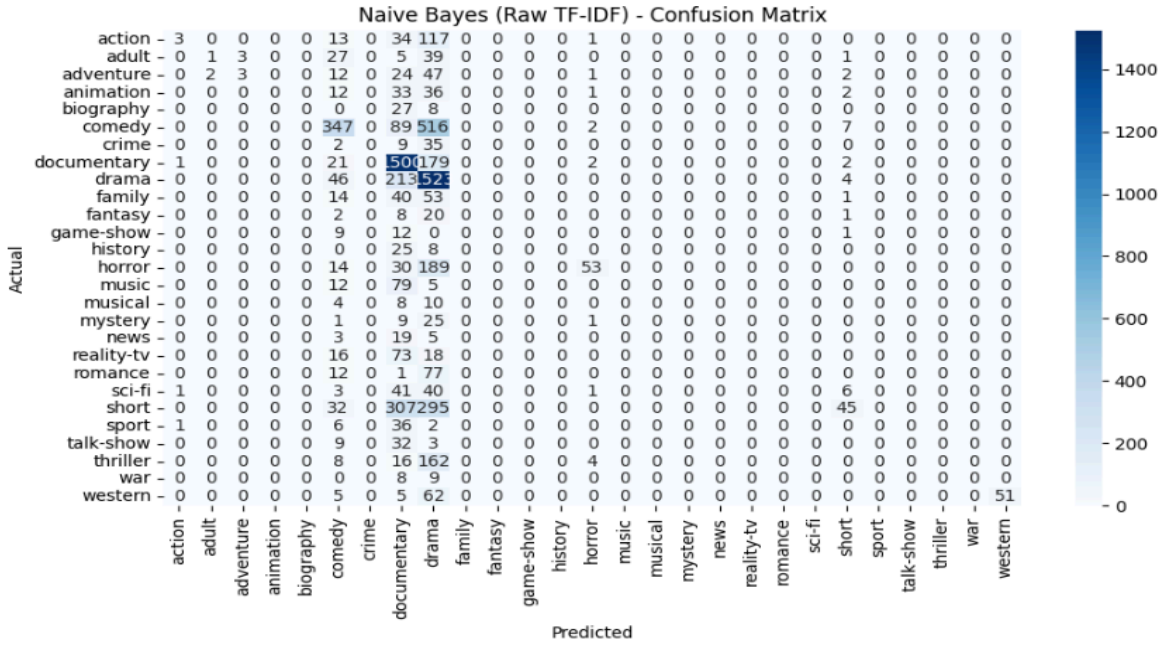


***Other confusion matrices present in the code file .**

4.2 Multinomial Naive Bayes:

Multinomial Naive Bayes is a fast, probabilistic classifier that assumes features are conditionally independent given the class label. It's well-suited for text classification problems using word frequency or TF-IDF features. In this task, it was applied to raw TF-IDF vectors and achieved a test accuracy of 50.37%, performing reasonably well on genres like Documentary, Drama, and Horror that have consistent and distinctive vocabulary.

However, the model struggled with low-frequency or semantically overlapping genres such as Biography, Game-Show, and Animation, often predicting dominant classes instead. The independence assumption and lack of contextual understanding limit its expressiveness in a dataset with subtle genre boundaries. Overall, it serves as a lightweight baseline but is outperformed by more expressive models.



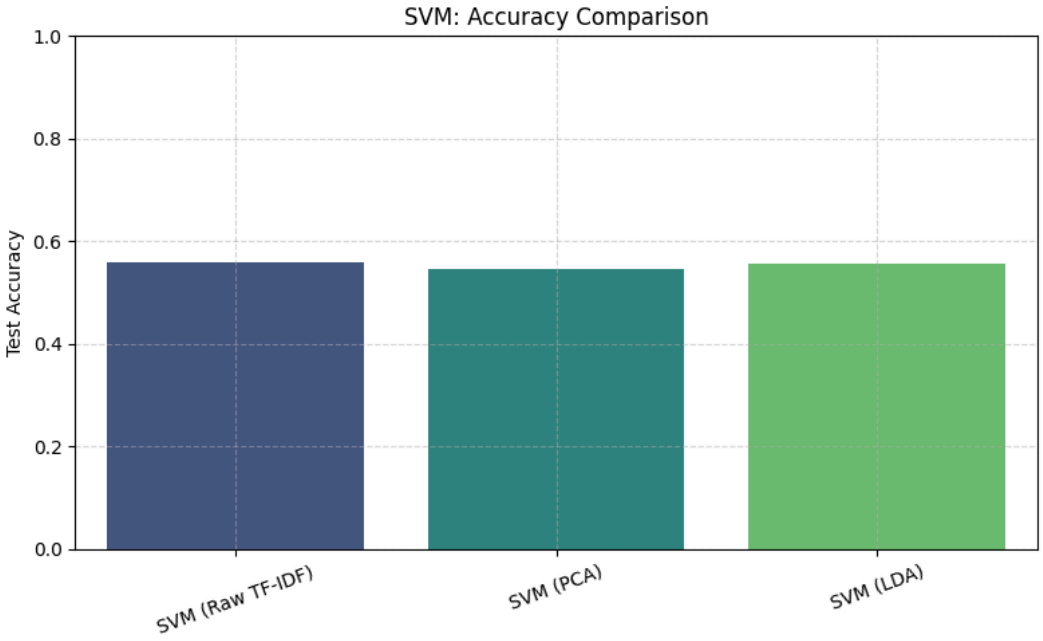
***Other confusion matrices present in the code file .**

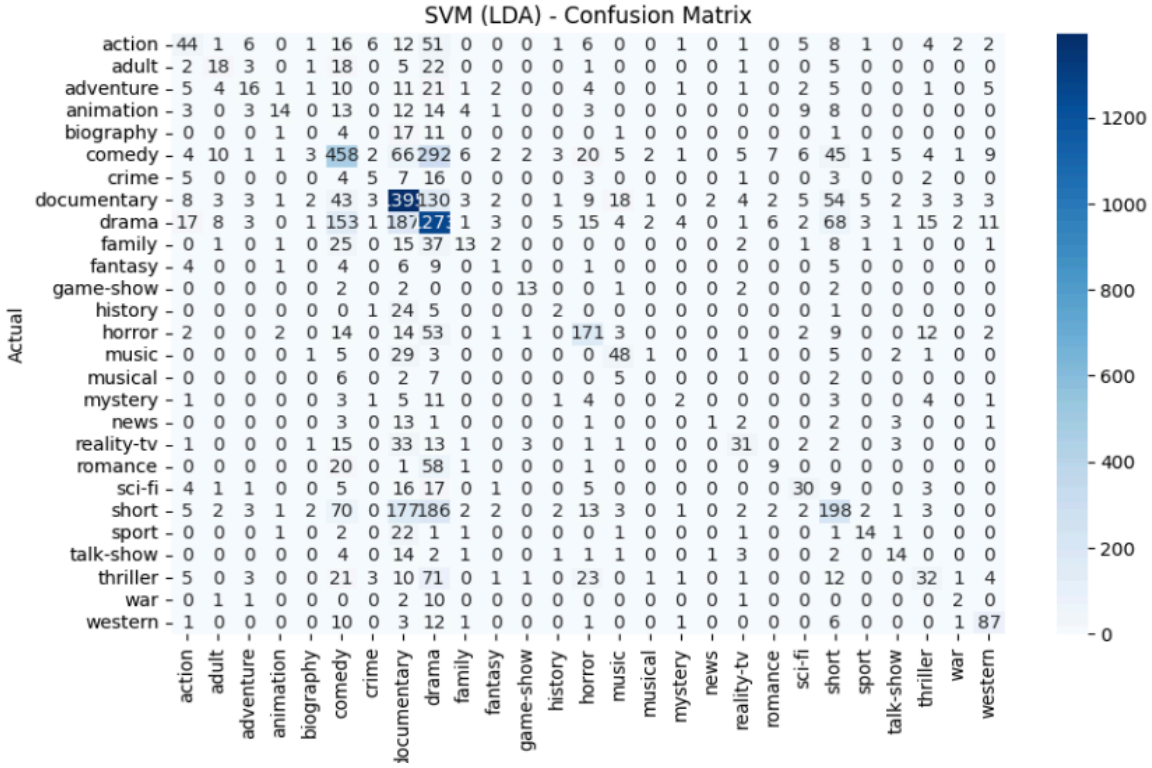
4.3 Linear SVM:

Support Vector Machines are powerful margin-based classifiers that perform well in high-dimensional spaces, such as text classification with sparse TF-IDF features. They are particularly effective in separating classes with complex decision boundaries while maintaining generalization through regularization. For our genre classification task, we used linear SVMs across raw, PCA-transformed, and LDA-transformed feature spaces.

Using raw TF-IDF vectors, the SVM achieved a respectable test accuracy of 55.79%, with a cross-validation score of 54.86% ± 0.0030, showing good generalization. It performed well on dominant genres like Drama, Comedy, and Documentary, but struggled with extremely underrepresented genres like Biography, Musical, and News, often assigning them zero predictions. This can be attributed to class imbalance and overlapping vocabulary.

Interestingly, the LDA-transformed feature space led to a slight increase in test accuracy (55.59%) and a notable jump in CV performance (63.55%), suggesting that supervised dimensionality reduction improved class separation. However, PCA led to a drop in both metrics. This reinforces that SVMs benefit more from class-informed projections (LDA) than unsupervised compression (PCA) when used on textual data with multi-class setups



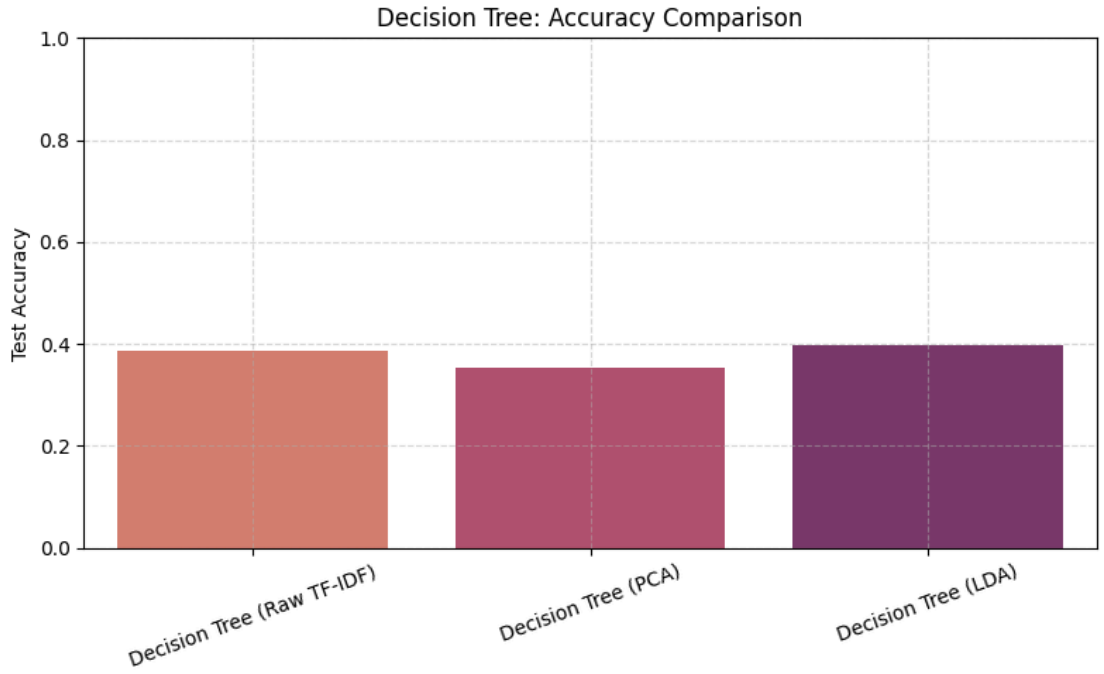


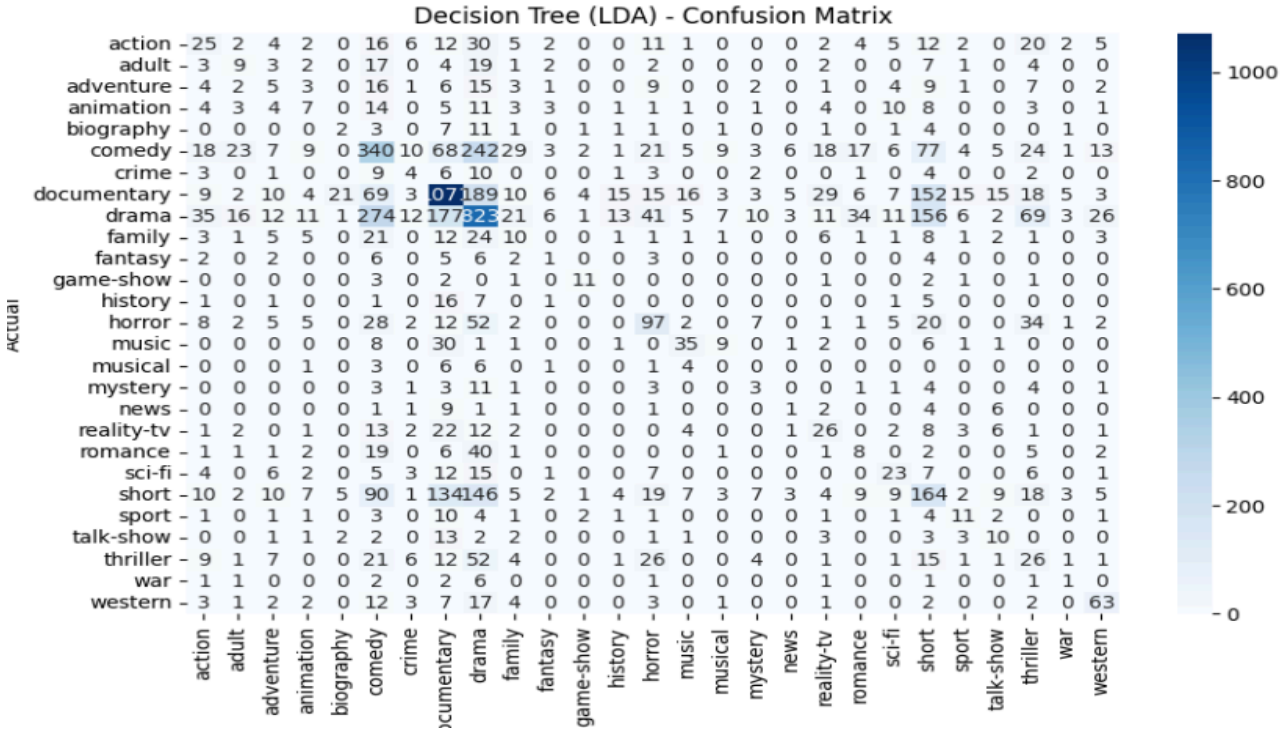
***Other matrices and details provided in code.**

4.4 Decision Tree:

Decision Trees are interpretable models that split data hierarchically based on feature thresholds. However, they tend to overfit, especially in high-dimensional, sparse datasets like raw TF-IDF vectors. This was evident in our results, where Decision Trees achieved a test accuracy of just 0.3870 and a cross-validation score of 0.3855 ± 0.0054, the lowest among all classifiers. The model struggled significantly with underrepresented genres like Biography, News, and Musical, often predicting only the dominant labels such as Drama or Documentary, resulting in very low macro and weighted F1-scores.

With PCA applied, performance further dropped to 0.3534, highlighting the model's reliance on specific sparse word indicators that were lost in projection. LDA provided a relative improvement, increasing test accuracy to 0.3966 and CV accuracy to 0.4295 ± 0.0050, indicating better class separation in the reduced space. However, the model still lacked robustness for generalization. Overall, while Decision Trees offer transparency, their poor performance on this 27-class problem shows they are not well-suited for such high-dimensional NLP tasks unless combined with more advanced ensemble techniques or balanced data strategies.



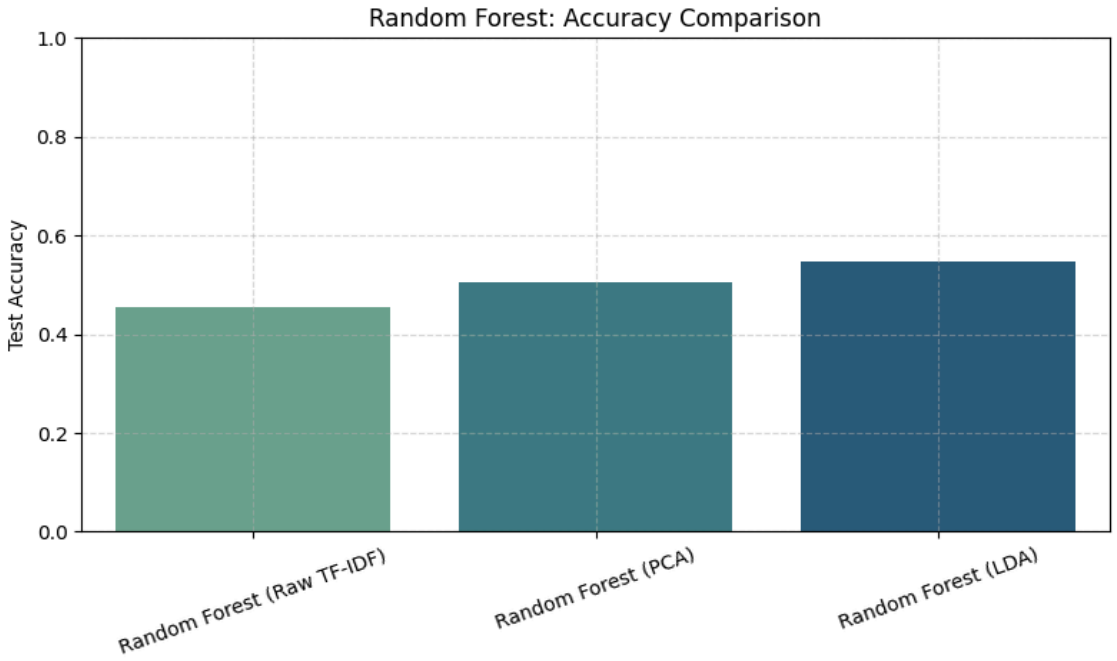


***Other details provided in code**

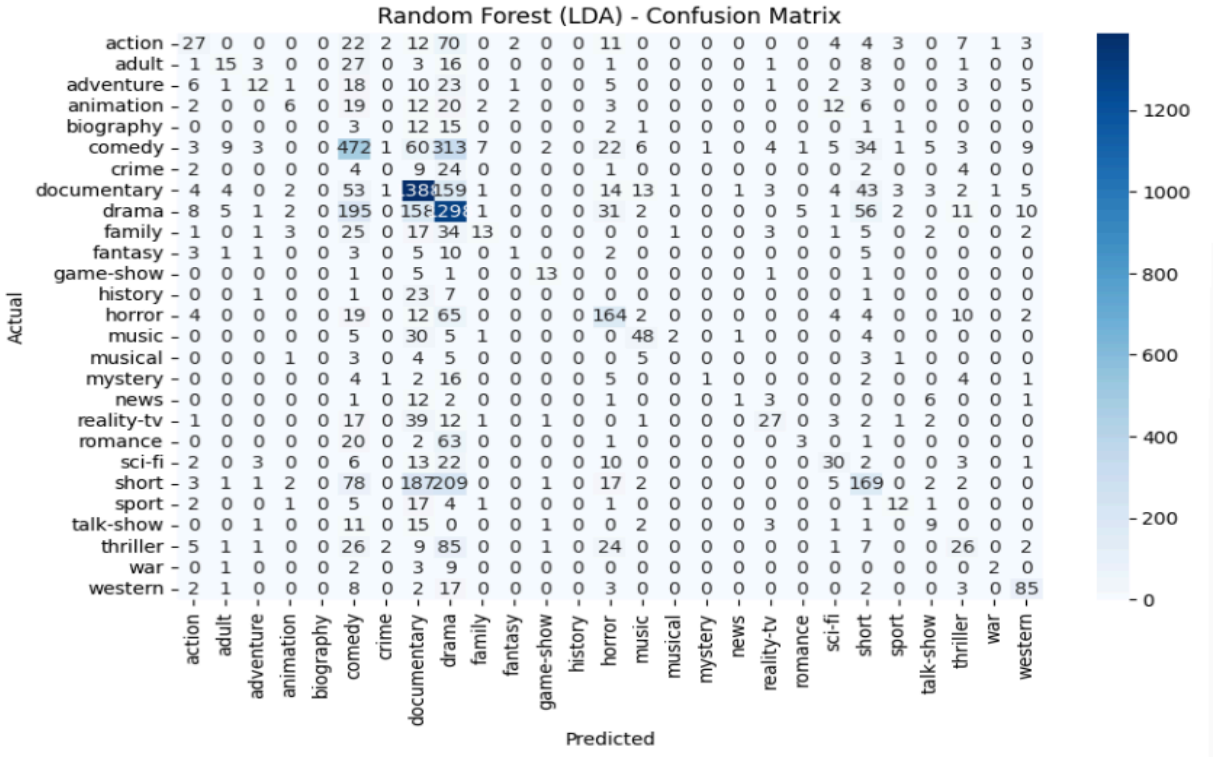
4.5 Random Forest:

Random Forest is an ensemble-based decision tree classifier that reduces variance by aggregating predictions across multiple decorrelated trees. On the raw TF-IDF vectors, its performance was notably weaker than other classifiers, with a test accuracy of 0.4549 and a CV mean of 0.4513 ± 0.0036. The model frequently overfitted dominant classes like Drama and Documentary while failing to learn from sparse, high-dimensional feature space especially for underrepresented genres such as Adult, Biography, and History, which often saw zero precision or recall. This is partially due to Random Forests not being well-suited for sparse or high-dimensional data without dimensionality reduction.

Upon applying PCA, the accuracy improved to 0.5041, and further to 0.5460 with LDA, where the supervised projection helped highlight discriminative features between classes. The LDA-transformed features aligned better with decision boundary learning, as seen in the boost to 0.5824 ± 0.0029 CV accuracy. Nonetheless, class imbalance continued to hinder performance on low-frequency genres. While Random Forest benefited from reduced feature spaces, its overall utility in genre classification remained limited compared to models like SVM and XGBoost, especially in handling fine-grained distinctions across the 27-genre space.



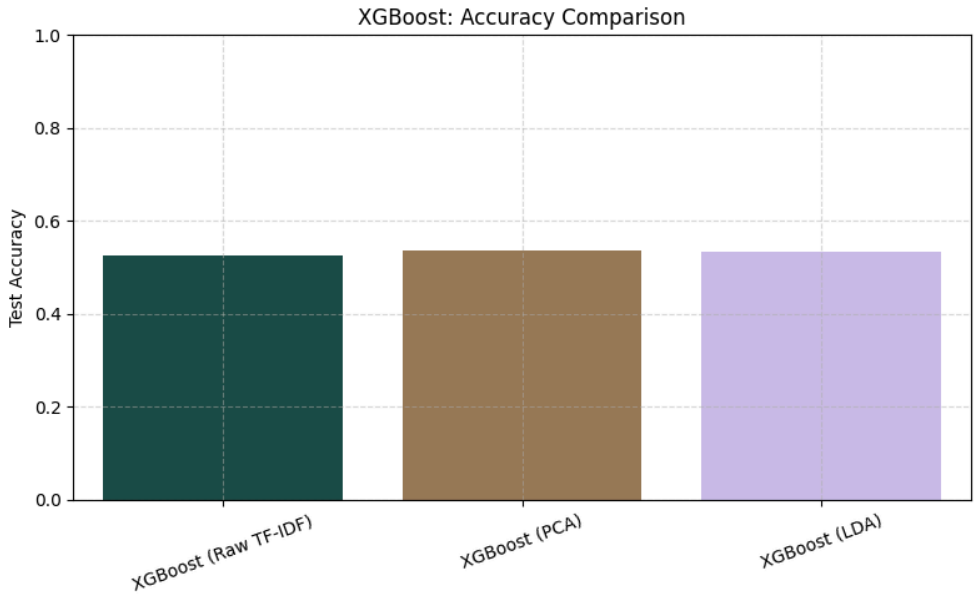
***Other details provided in code**

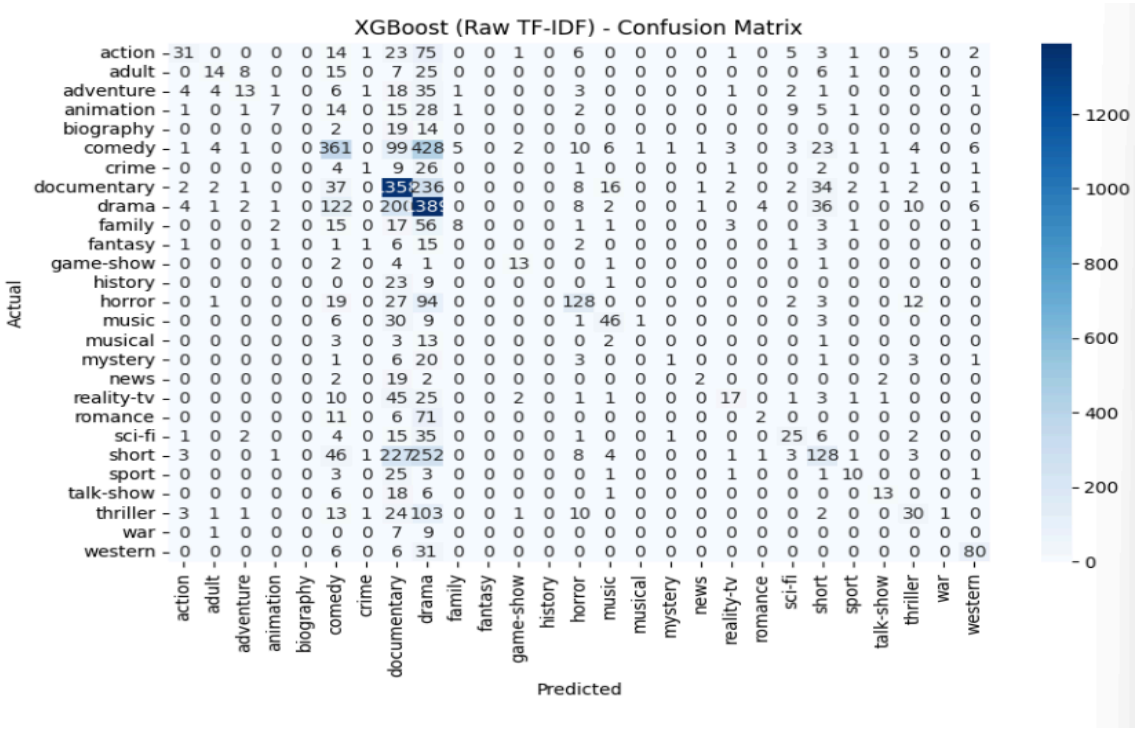


4.6 XGBoost:

XGBoost is a gradient boosting framework known for its regularization and scalability. It constructs an ensemble of weak decision tree learners, optimizing them using gradient descent to minimize a loss function. On the raw TF-IDF features, XGBoost achieved a test accuracy of 0.5253, with a cross-validation (CV) mean of 0.5183 ± 0.0037, indicating consistent but moderate generalization. It performed decently on major classes like Drama, Documentary, and Comedy, but struggled with minority and ambiguous genres such as Biography, Fantasy, and Musical. Its ability to capture nonlinear patterns helped slightly outperform simpler linear models on some genre boundaries, but it also overfitted certain rare genres, leading to inflated precision and poor recall for a few classes.

Upon applying dimensionality reduction, PCA marginally improved the test accuracy to 0.5357, while LDA yielded a nearly equivalent score of 0.5349. Notably, the cross-validation score with LDA rose significantly to 0.5832 ± 0.0043, suggesting that supervised dimensionality reduction aligns well with XGBoost's ensemble nature, possibly by emphasizing class-separating components. Despite this, performance across underrepresented genres remained weak, partly due to class imbalance and limited interpretability of tree splits in such high dimensional sparse spaces. Still, XGBoost demonstrated reasonable robustness across all settings, particularly when paired with LDA. These results underline its versatility as a strong traditional learner for multiclass NLP classification.



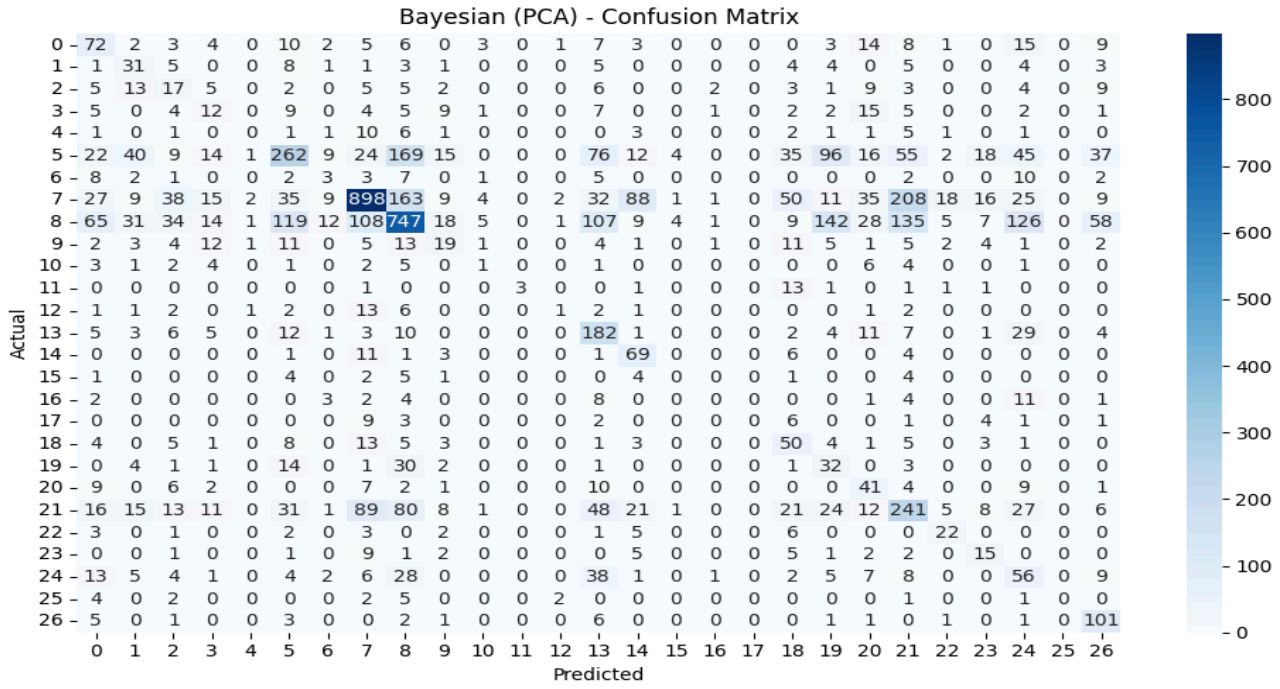


***Other data provided in code**.

4.7 Custom Bayesian Classifier:

The custom Bayesian classifier implemented here follows the Maximum Posteriori principle, estimating the most probable class given the input features by modeling each class as a multivariate Gaussian distribution. This approach computes class priors, mean vectors, and covariance matrices from training data, and applies Bayes' rule to assign class labels based on posterior probabilities. Dimensionality reduction using PCA or LDA was necessary due to numerical instability with full TF-IDF vectors.

On PCA-reduced features, the model achieved 41.07% test accuracy. It showed acceptable performance in high-frequency genres like Drama, Documentary, and Horror, but struggled with niche or ambiguous classes due to oversimplified Gaussian assumptions. With LDA features, accuracy improved to 50.27%, showing that supervised dimensionality reduction helps align the classifier with class-specific feature distributions. However, performance remains limited compared to SVM or ensemble models, likely due to high class overlap and assumptions of normality not holding in sparse text data.

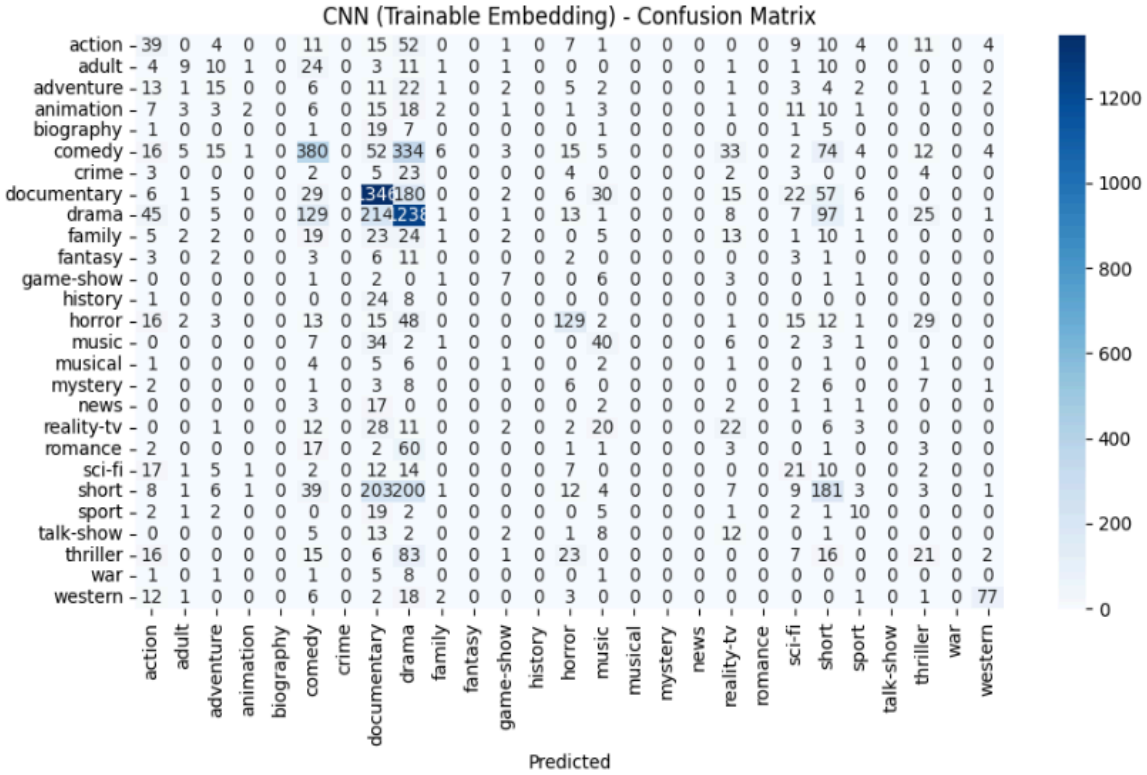


***Other data provided in code**.

4.8 Convolutional Neural Network (CNN):

Convolutional Neural Networks are well-suited for extracting local patterns in sequential data such as text, especially when represented as word or character embeddings. In this project, the CNN model achieved a test accuracy of 0.505, showing moderate performance in a complex 27-class setting. It performed relatively better in identifying high-frequency genres like Drama, Documentary, and Comedy, where patterns in descriptions are more consistent. Genres such as Horror and Music also saw decent f1-scores, suggesting that CNN was able to capture repeated phrasing or thematic cues within those categories.

However, the model underperformed on niche or low-resource genres such as Biography, Musical, News, and Fantasy, often defaulting to more common labels. Its macro-averaged F1-score was just 0.19, reflecting the model's struggle with minority classes. While the CNN benefited from its ability to learn hierarchical features, its effectiveness was limited by the imbalance in class representation and the relatively short length of descriptions in the IMDb dataset. Further improvement could involve data augmentation or pre-trained embeddings like GloVe or BERT-based input.

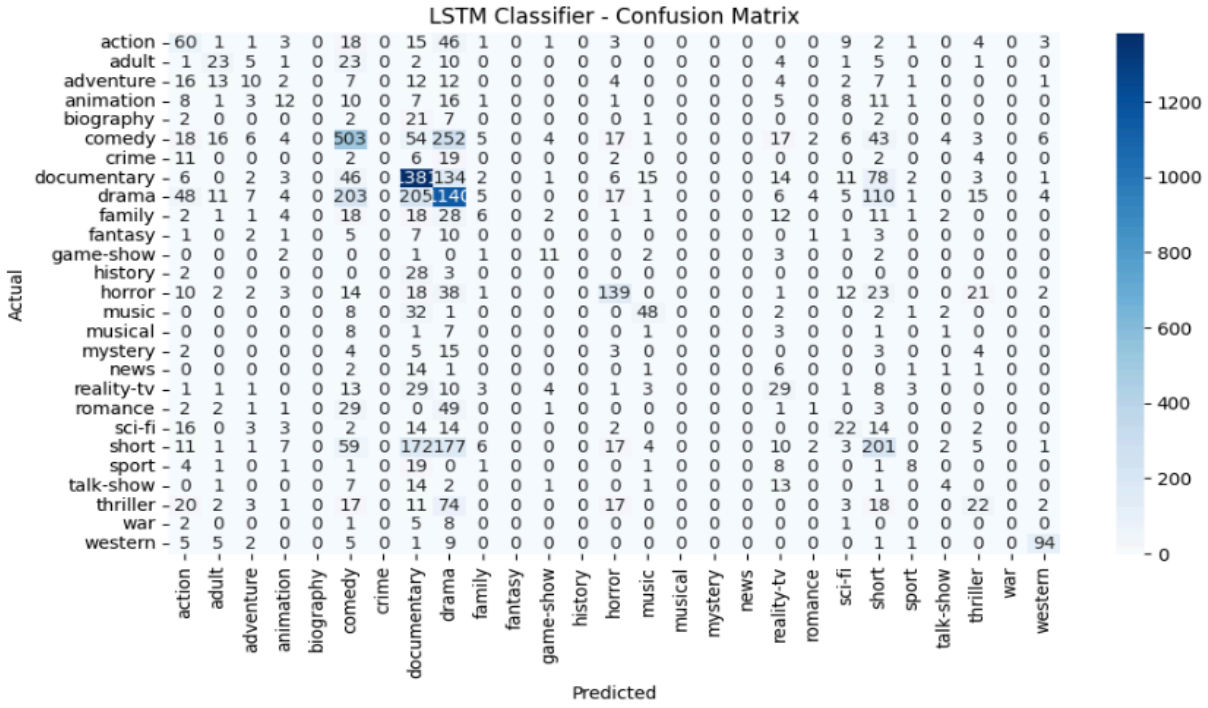


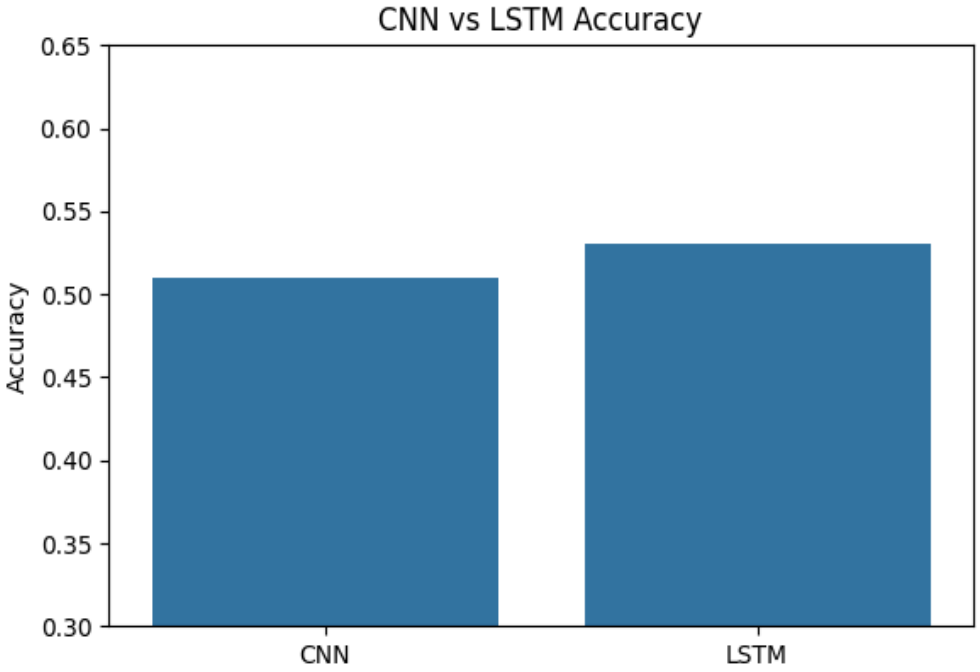
***Other data provided in code**.

4.9 Long Short-Term Memory (LSTM):

Long Short-Term Memory networks are designed to capture long-range dependencies in sequential data. Applied to the task of genre classification, the LSTM achieved a test accuracy of 0.531, slightly outperforming the CNN. It performed reliably on high-frequency genres like Drama, Documentary, and Comedy, as well as on Horror and Music, where temporal patterns or sequential word cues likely influenced predictions. Genres with regular phrasing in descriptions appeared to benefit from LSTM's sequential modeling capacity.

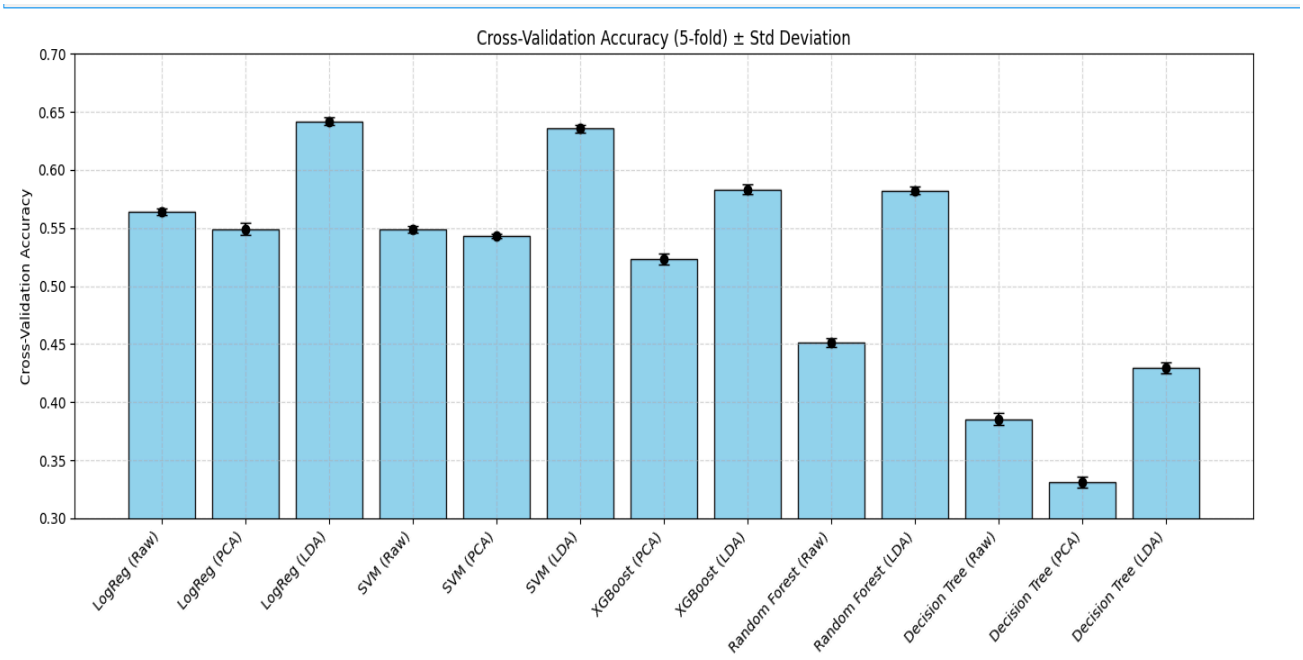
Despite this, the model struggled with minority or overlapping classes such as Biography, Musical, News, and Crime, often yielding zero precision and recall. The macro-average F1-score was 0.24, indicating that while the model excelled in dominant classes, its ability to generalize across all 27 classes was limited. Improvements could involve using pre-trained embeddings (like GloVe or FastText), increasing the training data, or applying attention mechanisms to help the model focus on genre-relevant phrases in the text.


LSTM Classifier - Confusion Matrix
Actual
Predicted
action
adult
adventure
animation
biography
comedy
crime
documentary
drama
family
fantasy
game-show
history
horror
music
musical
mystery
news
reality-tv
romance
sci-fi
short
sport
talk-show
thriller
war
western
1200
1000
800
600
400
200
0


CNN vs LSTM Accuracy
Accuracy
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0.30
CNN
LSTM

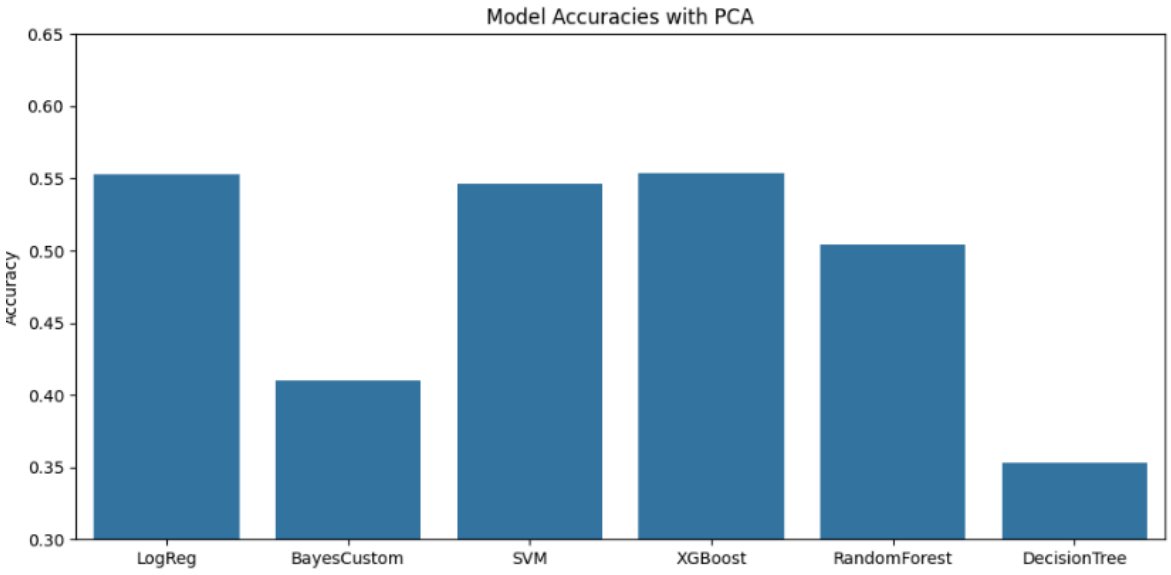
5. Cross-Validation & Variance:

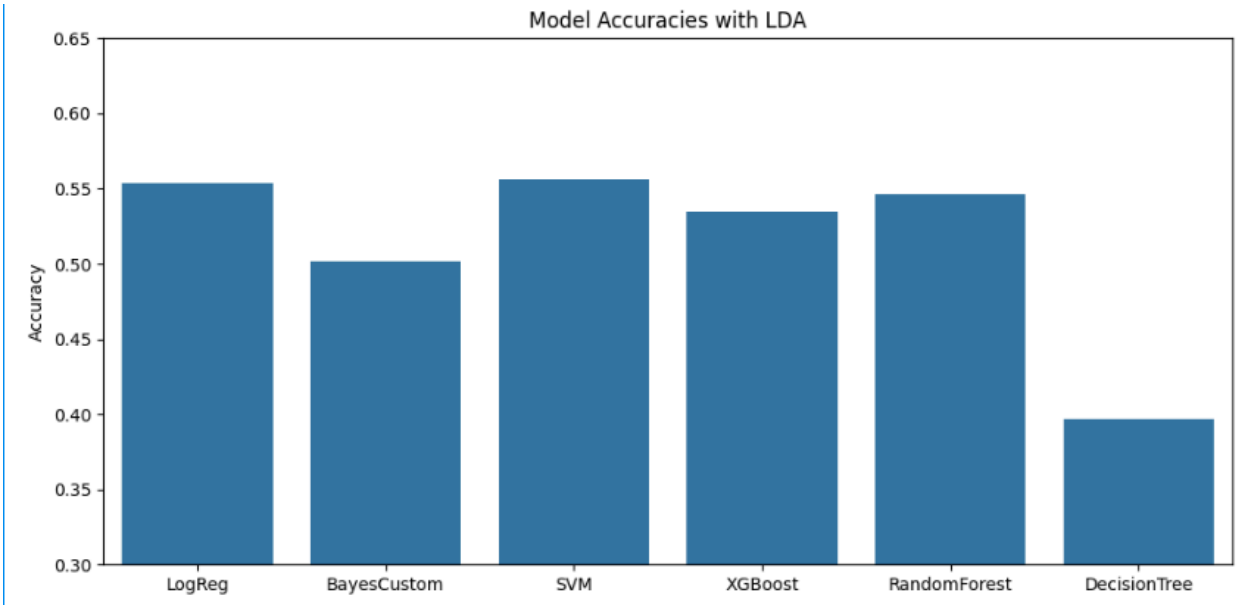
To evaluate the consistency of each classifier, 5-fold stratified cross-validation was applied to all traditional machine learning models. Logistic Regression and SVM exhibited strong stability, with low standard deviations (typically below 0.005), suggesting consistent performance across different folds. XGBoost and Random Forest also maintained moderate variance, while Decision Tree showed higher instability with deviations around ±0.0054. Deep learning models like CNN and LSTM were evaluated only on the test set due to computational constraints, but their performance remained competitive in comparison.



6. Dimensionality Reduction:

To manage the high dimensionality of TF-IDF features and improve generalization, two dimensionality reduction techniques were used. Principal Component Analysis (PCA) reduced the 2000-dimensional TF-IDF space to 100 components, preserving most of the variance while enabling faster training. Linear Discriminant Analysis (LDA), being supervised, reduced the features to c-1 components (number of classes minus one) and focused on maximizing inter-class separation. Visualizations revealed that LDA offered more distinct class clusters compared to PCA, particularly benefiting classifiers like SVM and Logistic Regression.

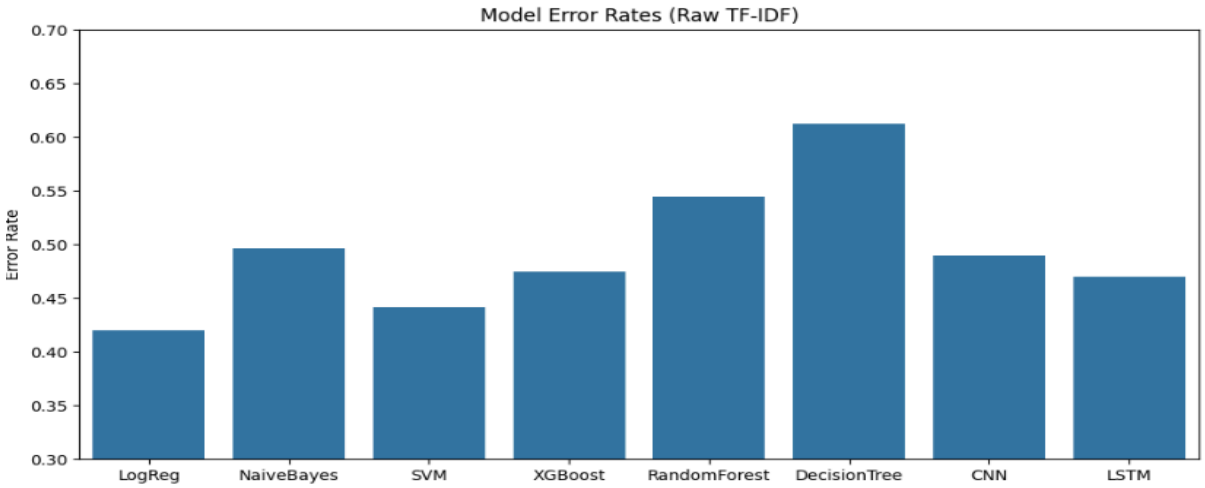




7. Error and Imbalance Analysis:

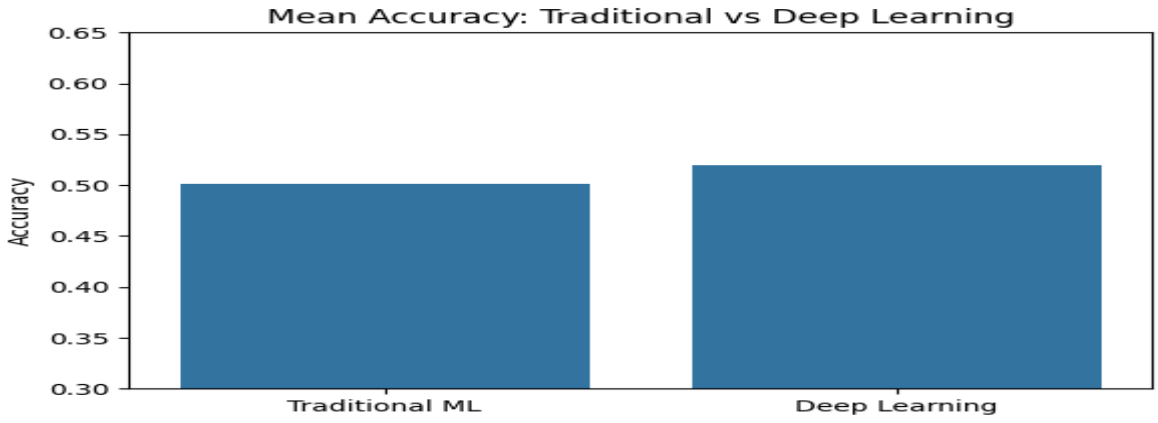
The dataset suffered from significant class imbalance, with genres like Drama and Documentary heavily represented, while classes like Musical, Talk-Show, and Game-Show had very few samples. This led to poor recall for underrepresented genres, with Talk-Show and Biography often misclassified. The Adult genre frequently overlapped with Drama and Crime due to shared vocabulary in descriptions. In contrast, genres like Horror and Fantasy showed relatively higher precision, likely due to the presence of distinctive keywords. Classification reports were crucial

in identifying these disparities, highlighting the need for alternative metrics beyond overall accuracy.



8. Accuracy Chart Summary:

The test accuracy chart ranks the classifiers based on their performance on the unseen test set. Deep learning models like LSTM and CNN outperformed most traditional methods, with CNN achieving the highest individual accuracy. On average, deep learning models performed better than traditional models, showing stronger generalization on complex genre semantics. Among traditional classifiers, XGBoost and SVM (especially with LDA) delivered the most reliable results. Naive Bayes and Decision Tree classifiers consistently underperformed, particularly on imbalanced or ambiguous genres. The custom MAP-based Bayesian classifier, while conceptually rigorous, lagged behind due to covariance instability and lack of regularization. This summary helps visualize the trade-off between model complexity and predictive performance.



9. Result analysis:-

Our analysis showed that certain genre classes were frequently confused due to semantic overlap and limited data. Genres like Adult were often misclassified as Drama or Crime, while Talk-Show, Reality-TV, and Documentary shared overlapping language. Rare genres such as Musical, News, and Biography had low precision and recall due to their limited representation and ambiguous textual cues, which made them harder to distinguish.

Class imbalance had a major impact on performance. Majority classes like Drama, Documentary, and Comedy dominated predictions, while low-frequency classes were often ignored or predicted incorrectly. Although we used weighted metrics to account for imbalance, they couldn't fully compensate for the lack of examples in rare classes. This highlights the importance of applying targeted solutions such as data augmentation or re-weighting in future work.

Cross-validation results showed that Logistic Regression, CNN, and SVM with LDA had stable performance with low standard deviation across folds (±0.003). On the other hand, Decision Trees showed the highest variability (±0.007), suggesting sensitivity to noisy data and overfitting. LDA improved the consistency of many models by providing more class-discriminative features.

Overall, CNN and LSTM models achieved the highest test accuracy, showing the benefit of deep learning in capturing semantic and contextual cues. Among traditional models, SVM with LDA performed best due to its margin-based optimization and supervised dimensionality reduction. In contrast, simpler models like Naive Bayes and Decision Tree performed poorly, particularly on sparse or imbalanced data.

10. Conclusion:

This project explored and experimented with a wide range of models including traditional machine learning classifiers and deep learning techniques for the task of IMDb genre classification. By applying text preprocessing, TF-IDF feature extraction, dimensionality reduction , and model evaluation techniques, the project helped analyze the strengths and limitations of each classifier in a real-world multi-class setting. Through detailed performance metrics and visualizations, we gained valuable hands-on experience with practical model selection, error analysis, and feature engineering. Overall, the project provided meaningful practical insight into the concepts taught in CSE 802 and deepened my understanding of applied machine learning workflows and pattern recognition.

11. Future Work:

- Extend to multi-label classification to capture genre overlap.
- Use transformer models like BERT for richer text understanding.
- Apply text data augmentation to balance rare genres.
- Explore model ensembling for improved generalization.
- Build genre co-occurrence graphs to enhance prediction context.

References:-

●Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. EMNLP. Provides foundational insights into applying CNNs for text classification.

●Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level Convolutional Networks for Text Classification. NIPS. Explores deep architectures that operate directly on raw text.

●Data Set Link: https://www.kaggle.com/datasets/hijest/genre-classification-dataset-imdb

●Joachims, T. (1998). Text categorization with Support Vector Machines: Learning with many relevant features. European Conference on Machine Learning (ECML). This paper established the effectiveness of SVMs for high-dimensional text classification tasks.

●Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. ACM Computing Surveys (CSUR), 34(1), 1–47. A comprehensive survey of traditional machine learning techniques applied to text classification.