Comparative Analysis of Machine Learning Models for Predicting Movie Success

Abhijith Harikumar
School of Computing
Dublin City University
Dublin, Ireland
Student ID: 23269494
abhijith.harikumar2@mail.dcu.ie

Joseph Mathew
School of Computing
Dublin City University
Dublin, Ireland
Student ID: 23267632
joseph.mathew8@mail.dcu.ie

***Abstract*—In this study, we conduct a comprehensive comparison of various machine learning algorithms to identify the most effective model for predicting movie success. The algorithms evaluated include K-Nearest Neighbors (KNN), Decision Tree, Random Forest, Neural Network, XGBoost and Linear SVC (Support Vector Classifier). The dataset utilized contains 6,544 entries, featuring attributes such as vote average, vote count, budget, runtime, popularity, original language, and genres. Additionally, we engineered new features like vote_avg_count and log_budget to enhance model performance. Each model's performance was rigorously assessed using accuracy, precision, recall, F1 score, and the area under the Receiver Operating Characteristic (ROC) curve (AUC-ROC). Our analysis aims to provide insights into which machine learning model offers superior accuracy and reliability in predicting the commercial success of movies, thereby offering valuable guidance for stakeholders in the film industry. The findings of this research have potential applications in various domains, including movie production and marketing strategies.**

***Index Terms*—K-Nearest Neighbors (KNN), Decision Tree, Random Forest, Linear SVC (Support Vector Classifier), Neural Network (MLP - MultiLayer Perceptron), XGBoost (Extreme Gradient Boosting), Accuracy, precision, recall, F1 score, ROC curve.**

I. Introduction

The film industry, a multi-billion-dollar enterprise, continually seeks innovative methods to predict the success of movies. Accurate predictions can significantly impact decision-making processes, from budget allocations to marketing strategies, ultimately influencing a film's profitability. With the advent of big data and advanced analytics, machine learning (ML) has emerged as a powerful tool to forecast movie success by leveraging historical and real-time data.

Machine learning models can analyze a plethora of features associated with movies, including vote average, vote count, budget, runtime, popularity, original language, and genres. These features provide a comprehensive view of a movie's potential appeal and financial performance. However, selecting the appropriate ML algorithm to accurately predict movie success remains a critical challenge.

This study aims to compare the effectiveness of four prominent machine learning algorithms: K-Nearest Neighbors (KNN), Decision Tree, Random Forest, and Support Vector Machine (SVM). Each algorithm offers distinct advantages and limitations. KNN is known for its simplicity and effectiveness in pattern recognition, while Decision Tree provides interpretability and ease of use. Random Forest, an ensemble method, enhances predictive accuracy and robustness by aggregating multiple decision trees. SVM, on the other hand, excels in high-dimensional spaces and is effective for classification tasks.

Our methodology includes preprocessing the dataset to manage both numerical and categorical features, followed by training each model and assessing its performance using a range of metrics, including accuracy, precision, recall, F1 score, and the area under the Receiver Operating Characteristic (ROC) curve. Through this systematic comparison of algorithms, our goal is to determine the most reliable model for predicting movie success.

The significance of this project extends beyond theoretical interest. Accurate predictions of movie success can revolutionize various aspects of the film industry. Studios and production houses can allocate budgets more efficiently, ensuring that resources are invested in projects with the highest potential for success. Insights from predictive models can inform marketing campaigns, allowing for targeted advertising and promotional efforts that maximize audience engagement. By understanding the factors that contribute to a movie's success, stakeholders can mitigate risks associated with new releases and make more informed decisions about project development. Furthermore, data-driven insights can guide content creators in developing storylines and genres that resonate with audiences, potentially leading to higher box office returns.

Additionally, the results of this research will offer valuable insights into the strengths and weaknesses of each algorithm, guiding stakeholders in the film industry towards more informed and data-driven decisions. The findings will also contribute to the broader field of predictive analytics by highlighting best practices for model selection in similar domains.

By demonstrating the practical applications and benefits of machine learning in the film industry, this study underscores the transformative potential of predictive analytics. It aims to bridge the gap between data science and industry practices, fostering a more innovative and efficient approach to filmmaking. Through this research, we hope to empower industry professionals with the tools and knowledge to leverage machine learning for competitive advantage, ultimately contributing to the advancement of both the film industry and the field of machine learning.

II. Related Works

A. Data Mining

Ahmad et al. created a reliable prediction model, a number of variables are carefully considered, including budget, actors, director, producer, set locations, story writer, movie release day, competing movie releases at the same time, music, release location and target audience [1]. The model's objective is to provide stakeholders with a reliable movie success forecast, reducing investment risks and efficiently directing resource allocation. Stakeholders are able to make well-informed decisions in the ever-changing film industry by using the insightful information that data mining provides. Their work is advantageous because it identified strong correlations between various criteria and a movie's success rating, enabling predictions even before the movie is released. In this work, a mathematical model for determining the success rating of upcoming films based on specific parameters was developed. The mathematical model led to the conclusion that one factor influencing a movie's success rating was its genre. It was also established that the actors in the films had an impact on their success. Actors and genres were shown to be strongly correlated, suggesting that some Actors are more likely to work in particular genres. This work's present restriction is that it only looks at Bollywood films.

Sentiment PLSA (S-PLSA) was used in the study conducted by Xiaohui et al. to capture the complex sentiments conveyed in reviews [2]; ARSA, an Autoregressive Sentiment-Aware model for sales prediction, is then developed. Furthermore, the ARSQA model is proposed, which integrates review quality prediction with sales performance forecasting. The study highlights how crucial it is to comprehend the opinions expressed in Internet reviews in order to provide decision-makers with useful information. The study offers a thorough method for comprehending and applying internet reviews for sales prediction by using S-PLSA to summarize sentiment data and adding it into predictive models. The efficacy and precision of the suggested models are validated by comprehensive trials carried out on film datasets, providing significant perspectives for enterprises looking to utilize the prognostic potential of online reviews for strategic choices. The study highlights the S-PLSA model's adaptability and suggests uses for it beyond sales forecasting, like sentiment-based review grouping and classification. Prospective avenues for investigation encompass examining patterns in sentiment expression on the Internet and creating increasingly advanced algorithms for sentiment analysis and assessing review quality. In conclusion, this study advances knowledge about sentiment analysis in online reviews and how it affects forecasts of product sales performance. It gives companies a platform for efficiently utilizing the predictive potential of internet evaluations.

Olubukola et al. [3] indicated that the problem facing the film business, es pecially Nollywood in Nigeria, is that many films are made, but very few of them are successful at the box office. In order to tackle this issue, they proposed a movie success prediction model that makes use of machine learning algorithms and data mining approaches. The model attempts to predict the success or failure of upcoming films by examining factors like marketing budgets, compe tition, release dates, and actor history data. IMDb Metadata was subjected to data mining techniques, which ensured data quality through integration and cleaning procedures [3]. The prediction model includes learning techniques and data reading that are essential for analysis and assessment. Methods such as feature reduction and data pre-treatment were applied to control the dataset's complexity and improve accuracy. The dataset was divided into train and test sets using a decision tree technique, producing acceptable precision and accu racy metrics. This made it possible for interested parties to forecast the genres of films and decide on appropriate production tactics. This model is important because it can be used to change the parameters for movies, which will raise the likelihood of blockbusters and help viewers assess the calibre of upcoming movies. A trustworthy prediction system is necessary for the film industry to succeed, as audience acceptance is crucial in the cutthroat movie business.

Galvo et al. [4] suggested a study aiming to predict movie profits by constructing a predictive model using various data mining techniques, including neural networks, regression, and decision trees. The model predicts box office revenue by utilizing three different approaches for the dependent variable: interval, categorical, and binary. The objective is to analyze the differences and predictive influences of each approach. Two metrics are employed to assess prediction accuracy: misclassification error for categorical models and average squared error for continuous models. The study concludes that the multi-layer perceptron yields the best predictive results [4]. However, the multi-class model exhibits a significantly higher error rate compared to the others, attributed to the increased complexity resulting from predicting multiple classes.

B. Machine Learning

Tanishq et al., investigated the increasing number of films worldwide, particularly in India, and the challenge of predicting the success of Bollywood films [5]. In the recent past, Machine Learning calculations have been used to recognize fascinating examples from volumes of data and help the dynamic procedure in business conditions. The achievement pace of a film is greatly dependent on the marketing strategies the producers use on social media. They propose a method that combines machine learning and natural language processing to forecast a film's box office performance based on social media buzz. By analyzing sentiment on platforms like YouTube and Twitter, considering likes, dislikes, comments, and tweets, along with the star power of actors and actresses, they aim to assist stakeholders in the film industry with forecasting and managing a film's box office performance and screen distribution.

Shreehar et al.'s study addresses a classification problem aiming to predict a movie's success based on various features [6]. Two movie datasets, along with features obtained through web scraping, are used to create training and testing datasets. Four Machine Learning classifiers, namely Stochastic Gradient Descent, Random Forests, LinearSVC, and Extra Trees, are employed to classify these datasets. The study evaluates the performance metrics of these classifiers on the movie datasets and draws conclusions based on the results. Random Forests and Extra Trees emerge as the most effective classifiers. Additionally, the study finds that the revenue feature adds noise to the system and does not contribute value to model fitting, while the "directors" feature significantly enhances model efficiency.

The forecasting problem is transformed into a classification problem by Sharda et al. [7], where movies are categorized into nine performance cate gories ranging from "flop" to "blockbuster." The goal of the approach is to help exhibitors, distributors, and studios make wise choices. The results show how well the neural network performs in comparison to other models; it can identify a movie's success category with a success rate of 36.9% and an accuracy of 75.2% within a single category [7]. They clain that the neural network model has the potential to predict not only box office success but also success rates for other media items. Entertainment companies can determine how changes to model parameters—like actors, release dates, or technical effects—will affect their bottom line.

Asad et al [8] investigated the understudied topic of grading pre-release film popularity according to intrinsic factors like budget, language, country, stars, and directors. This work presents a classification scheme employing C4.5 and PART classifier algorithms, in contrast to previous research that mostly focuses on binary classification or recommendation systems. It also looks at the relationship between post-release film qualities. The new method models the classification process by combining user-determined movie ratings with intrinsic features. The relationship between the initial budget and financial returns after release is clarified by experimental findings. Findings show that director rank and budget have a significant influence on how popular pre-release films are ranked. On the other hand, when examining post-release movie datasets, the classifiers exhibit limitations. Higher budgets are correlated with higher financial returns and vice versa, according to correlation coefficients between budget and financial returns (domestic, international, and foreign) [8]. When making decisions about movie rentals, streaming services, and brand sponsorships, producers, directors, and film finance organizations can all profit practically from the suggested model and machine learning steps.

In comparison to other studies, Du et al. presented improvements in feature extraction techniques and prediction models as it investigates the use of microblogs for box office success prediction [9]. It explores two primary areas: content-based features utilizing a novel box office-oriented semantic classification method, and count-based features that focus on user factors to filter out irrelevant microblogs. In order to improve forecast accuracy and dependability, the study makes use of more sophisticated machine learning models, such as SVM and neural networks. [9]The study assesses how well the suggested methods work in comparison to conventional prediction techniques and earlier microblog prediction approaches. The findings exhibit exceptional precision and dependability, particularly in forecasting consequences associated with user actions on social media networks such as ticket sales. Nonetheless, the research highlights constraints in forecasting results unconnected to social media such as election results. The report concludes by outlining a number of directions for more investigation. Priority one for practical applications is to determine what kinds of consequences may be predicted from microblog data. Furthermore, investigating techniques for microblog-based real-time event prediction presents an interesting challenge. Making use of the connections between users in intricate social networks may help improve the precision and adaptability of prediction techniques. Lastly, adding latent semantic analysis (LSA) to the box office-focused classification technique may enhance prediction power even further.

The internet has become the premier source of information, with every field extensively uploading data and becoming increasingly efficient. The film industry is no exception, producing numerous movies rapidly, prompting producers to seek early predictions of a movie's success. This study by Syed et al. [10] explores the hidden patterns contributing to a movie's popularity using statistical and machine learning algorithms. While previous research focused on machine learning techniques applied to various sources like blog articles and social media for success prediction, it often lacked depth in ongoing movie data analysis. This paper investigates the relevance of such data for predictive purposes. Using publicly available data from the Internet Movie Database (IMDb), five machine learning algorithms were implemented: Generalized Linear Model (GLM), Deep Learning (DL), Decision Tree (DT), Random Forest (RF), and Gradient Boosted Tree (GBT), with Root Mean Squared Error (RMSE) as the performance metric [10]. The study found that GLM was the top-performing regression classifier, achieving 47.9% accuracy due to its lower RMSE value [10].

Our study distinguishes itself by employing a comprehensive suite of machine learning algorithms, including Random Forest, Linear SVC, KNN, Decision Tree, MLP Neural Network, and XGBoost, with rigorous hyperparameter tuning to optimize predictive accuracy. Unlike previous works that often focused on specific algorithms or limited features, we extensively applied feature engineering and meticulously assessed its impact on model performance. Moreover, our research uniquely examined the effects of various training-test split ratios on model stability and performance, offering a nuanced understanding of predictive influences across different machine learning techniques. This multifaceted approach facilitated a thorough comparison and validation of model effectiveness in predicting movie success, thereby advancing the field of predictive modeling in the film industry.

III. Dataset Description and Preprocessing

The dataset utilized in this study is sourced from the TMDB (The Movie Database) Movies Dataset, which initially contained a wide range of columns encompassing various aspects of movie metadata. After cleaning and preprocessing, the dataset comprised 6544 entries, providing a substantial foundation for our analysis.

A. Handling Missing Values

Upon examining the dataset, we identified several columns with missing values. The columns with null values included backdrop_path, homepage, imdb_id, overview, poster_path, tagline, production_companies, production_countries, spoken_languages, and keywords. To ensure data quality and consistency, we decided to remove some columns that were deemed unnecessary for our analysis. These columns were backdrop_path, homepage, poster_path, original_title, and tagline.

For the remaining columns with missing values, we implemented specific strategies to handle these gaps. The keywords column, which provides tags relevant to the movie's content, had its null values replaced with the placeholder 'No Keywords'. Similarly, the overview column, which offers a brief description of the movie, had missing entries replaced with 'No Overview'. The columns production_companies, production_countries, and spoken_languages, all critical for understanding the movie's production background and linguistic aspects, had their null values replaced with 'Other'.

B. Feature Engineering

To enhance the dataset's predictive power, we conducted feature engineering and created several new columns including vote_average_count, budget_popularity_interaction, log_budget and popularity_bin. These features were designed to capture more nuanced relationships within the data.

However, upon calculating the Variance Inflation Factor (VIF), we observed that popularity_bin and vote_average_count increased the multicollinearity of popularity and vote_count, respectively. High multicollinearity can undermine the reliability of regression analyses by inflating standard errors. Therefore, to address this issue, we decided to drop the engineered columns popularity_bin and vote_average_count. The VIF values for the remaining features were as follows:

No	Variable	VIF
1	vote_average	1.198549
2	vote_count	1.743658
3	budget	2.187939
4	runtime	1.766640
5	popularity	2.464193
6	budget_popularity_interaction	2.745502
7	log_budget	2.169345

These VIF values, all below the threshold of 5, indicate minimal intercorrelations among the variables, affirming that each variable provides unique information to our models.

C. Data Cleaning and Preparation

The thorough cleaning and preprocessing steps ensured that the dataset was complete and ready for subsequent analysis. The final dataset, consisting of 6544 entries, provided a robust foundation for feature engineering and model training processes aimed at predicting movie success. This detailed approach to data preparation underscores the importance of meticulous data handling in ensuring the efficacy of machine learning applications in real-world scenarios.

Through careful preprocessing, including handling missing values, removing unnecessary columns, and addressing multicollinearity, we transformed the dataset into a clean and structured form. This transformation was crucial in maintaining the integrity and usability of the data, allowing for more accurate and reliable results in our machine learning models.

IV. Methodology

In our project, we aimed to predict movie success using machine learning models such as Random Forest, Linear SVC, KNN, Decision Tree, Neural Network and XGBoost with an initial focus on the Random Forest algorithm. The dataset, sourced from the TMDB Movies Dataset, included 6,544 entries after comprehensive cleaning and preprocessing.

We introduced a new feature, Return on Investment (ROI), calculated as

$$ROI = revenue - budget/budget \tag{1}$$

Based on this, we created a binary target variable, Success, where movies with an ROI greater than 1 were labeled as successful.

The dataset utilized in this study comprises a comprehensive set of features including vote_average, vote_count, budget, runtime, popularity, original_language, genres, budget_popularity_interaction, and log_budget, with the target variable being Success. The preprocessing and feature engineering processes were meticulously designed to ensure optimal data preparation for model training and evaluation.

Initially, feature selection was carried out to identify both numeric and categorical variables, alongside the multi-label categorical variable genres. To handle the multi-label nature of the genres feature, a custom transformer, MultiLabelBinarizerTransformer, was developed. This transformer leverages the MultiLabelBinarizer to convert genres into multiple binary features, making them suitable for model input. The numeric features included vote_average, vote_count, budget, runtime, popularity, budget_popularity_interaction, and log_budget, while the categorical feature was original_language.

A ColumnTransformer was employed to manage preprocessing for numeric, categorical, and multi-label categorical data. Standardization of numeric features was conducted using StandardScaler to ensure that each feature contributes equally to the model's performance. Categorical data was transformed using OneHotEncoder, which converts categorical variables into a format that can be utilized by machine learning algorithms. The handle_unknown='ignore' parameter was employed to gracefully handle any unseen categories in the test data. The custom transformer for genres was integrated into the preprocessing pipeline to manage the multi-label nature of the genres feature.

To develop robust predictive models, various machine learning algorithms were employed, including Decision Tree, K-Nearest Neighbors (KNN), Multi-Layer Perceptron (MLP), Support Vector Classifier (SVC), Random Forest, and XGBoost classifiers. Each algorithm was integrated into a pipeline that included preprocessing, feature selection, and classification. For feature selection, SelectKBest with mutual_info_classif as the scoring function was used to reduce the number of features to 20, enhancing the model's efficiency and performance.

A. Random Forest

A Random Forest classifier was integrated, with hyperparameters including n_estimators, max_depth, min_samples_split, min_samples_leaf, and max_features. The RandomizedSearchCV method was used to tune these hyperparameters for optimal performance.

B. Linear SVC (Support Vector Classifier)

An SVC with a linear kernel was utilized, incorporating hyperparameters such as the regularization parameter C sampled from a log-uniform distribution between 0.001 and 1. The model's performance was evaluated using RandomizedSearchCV.

C. Decision Tree Classifier

A Decision Tree classifier from the scikit-learn library was used. Hyperparameter tuning was conducted using RandomizedSearchCV with a parameter distribution that included max_depth, min_samples_split, min_samples_leaf, and criterion.

D. KNN (K-Nearest Neighbors Classifier)

A KNN classifier was employed, with hyperparameters including n_neighbors, weights, and p (power parameter for the Minkowski metric). The tuning process involved RandomizedSearchCV to identify the optimal configuration.

E. Neural Network (MLP Classifier)

An MLP classifier was configured with a maximum of 1000 iterations to ensure convergence. The hyperparameter search included configurations for hidden_layer_sizes, activation, solver, alpha, and learning_rate.

F. XGBoost Classifier

An XGBoost classifier from the xgboost library was selected. Hyperparameters included n_estimators, max_depth, learning_rate, subsample, and colsample_bytree. The hyperparameter tuning process used RandomizedSearchCV to optimize these parameters.

G. Performance Evaluation

The performance of each model was evaluated using multiple metrics:

- F1 Score: Evaluated using f1_score to balance precision and recall
- Accuracy: Calculated using accuracy_score.
- Classification Report: Provided detailed performance metrics, including precision, recall, and F1 score for each class.
- Confusion Matrix: Generated using confusion_matrix to visualize the classifier's performance.
- ROC-AUC Score: Calculated using roc_auc_score to measure the model's ability to distinguish between classes.

Visualization techniques, including confusion matrices and ROC curves, were employed using seaborn and matplotlib to visually assess each model's performance. These comprehensive methodologies ensured a robust approach to preprocessing, model development, hyperparameter tuning, and evaluation, adhering to best practices in machine learning model development.

In summary, this study applied a rigorous methodology involving feature engineering, preprocessing, model development, and hyperparameter tuning across several machine learning algorithms. The use of advanced techniques and careful parameter optimization facilitated the development of robust predictive models, evaluated through a suite of performance metrics to ensure reliability and effectiveness in predicting the target variable, Success.

V. Results

The evaluation of various machine learning models, including Random Forest, Linear Support Vector Classifier (SVC), K-Nearest Neighbors (KNN), Decision Tree, Multi-Layer Perceptron (MLP) Neural Network, and XGBoost, provided a comprehensive understanding of their performance in predicting movie success.

A. Random Forest

The Random Forest model, Without feature engineering achieved an accuracy of approximately 74.9%, an F1 score of 0.671, and a ROC-AUC score of 0.7977. The classification report indicated a precision of 75% for class 0 and 75% for class 1, with recall values of 85% and 60%, respectively. The confusion matrix showed that 645 true negatives and 335 true positives were correctly identified.

With feature engineering, the model achieved a slightly lower accuracy of 74.3% and an F1 score of 0.662, with a comparable ROC-AUC score of 0.7982. The classification report indicated a precision of 74% for class 0 and 75% for class 1, with recall values of 85% and 59%, respectively. The confusion matrix showed that 644 true negatives and 329 true positives were correctly identified.

These results indicate that feature engineering did not significantly impact the performance of the Random Forest model, suggesting that the model already captured most of the relevant patterns in the data without additional features.

TABLE I
PERFORMANCE METRICS

Performance Metrics (without Feature Engineering)				
Classifier Type	**Accuracy**	**Precision**	**Recall**	**F_1 Score**
Random Forest	74.87%	75%	85%	67.07%
Linear SVC	70.36%	68%	93%	52.80%
KNN	71.81%	72%	85%	62.00%
Decision Tree	70.13%	70%	84%	59.00%
Neural Network	73.57%	74%	85%	65.19%
XGBoost	73.26%	73%	85%	64.72%
Performance Metrics (with Feature Engineering)				
Classifier Type	**Accuracy**	**Precision**	**Recall**	**F_1 Score**
Random Forest	74.33%	74%	85%	66.20%
Linear SVC	70.21%	68%	93%	53.01%
KNN	73.80%	72%	89%	63.00%
Decision Tree	70.89%	71%	84%	61.00%
Neural Network	73.64%	77%	77%	68.78%
XGBoost	71.43%	71%	84%	61.60%

B. Linear SVC

The Linear SVC model, without any feature engineering achieved an accuracy of 70.36% and an F1 score of 0.5280. The classification report indicated a precision of 68% for class 0 and 81% for class 1, with recall values of 93% and 39%, respectively. Later, We included additional engineered features such as budget_popularity_interaction and log_budget. The numeric features (vote_average, vote_count, budget, runtime, popularity, budget_popularity_interaction, and log_budget) were standardized, while categorical and multi-label features were processed similarly to the non-engineered approach. Here, the model achieved an accuracy of 70.21% and an F1 score of 0.5301. The classification report indicated a precision of 68% for class 0 and 80% for class 1, with recall values of 93% and 40%, respectively. The ROC-AUC score was 0.7666 for both models, indicating a good balance between true positive and false positive rates.

The Linear SVC model demonstrated consistent performance in predicting movie success, with an accuracy around 70% and an F1 score slightly over 53%, whether feature engineering was applied or not. The feature engineering introduced additional complexity but did not significantly improve the model's performance.

C. KNN

For the KNN classifier, without feature engineering, The model achieved an accuracy of 71.81% and an F1 score of 0.62. The classification report indicated a precision of 72% for both classes, with recall values of 85% and 55%, respectively. The confusion matrix showed 638 true negatives and 302 true positives correctly identified, and the ROC-AUC score was 0.7611. With feature engineering, With feature engineering, The model achieved an accuracy of 73.80% and an F1 score of 0.63. The classification report indicated a precision of 72% for class 0 and 78% for class 1, with recall values of 89% and 53%, respectively. The confusion matrix showed 674 true negatives and 292 true positives correctly identified, and the ROC-AUC score was 0.7672.

This improvement suggests that KNN, being a distance-based algorithm, benefits from the additional features that likely enhanced the discriminatory power of the input space.

D. Decision Tree

The Decision Tree classifier, without feature engineering, The model achieved an accuracy of 70.13% and an F1 score of 0.59. The classification report indicated a precision of 70% for both classes, with recall values of 84% and 52%, respectively. The confusion matrix showed 631 true negatives and 287 true positives correctly identified, and the ROC-AUC score was 0.7376. With feature engineering, The model achieved an accuracy of 70.89% and an F1 score of 0.61. The classification report indicated a precision of 71% for both classes, with recall values of 84% and 53%, respectively. The confusion matrix showed 635 true negatives and 293 true positives correctly identified, and the ROC-AUC score was 0.7503.

The feature engineering introduced additional complexity but only slightly improved the model's performance.

E. Neural Network (MLP Classifier)

The Neural Network (MLP) classifier demonstrated robust performance both with and without feature engineering. Without feature engineering, The model achieved an accuracy of 73.57% and an F1 score of 0.6519. The classification report indicated a precision of 74% for both classes, with recall values of 85% and 58%, respectively. The confusion matrix showed 639 true negatives and 324 true positives correctly identified, and the ROC-AUC score was 0.7827. Later, with feature engineering this model achieved an accuracy of 73.64% and an F1 score of 0.6878. The classification report indicated a precision of 77% for class 0 and 69% for class 1, with recall values of 77% and 69%, respectively. The confusion matrix showed 584 true negatives and 380 true positives correctly identified, and the ROC-AUC score was 0.7981.

The MLP classifier demonstrated robust performance in predicting movie success, with an accuracy around 73% and an F1

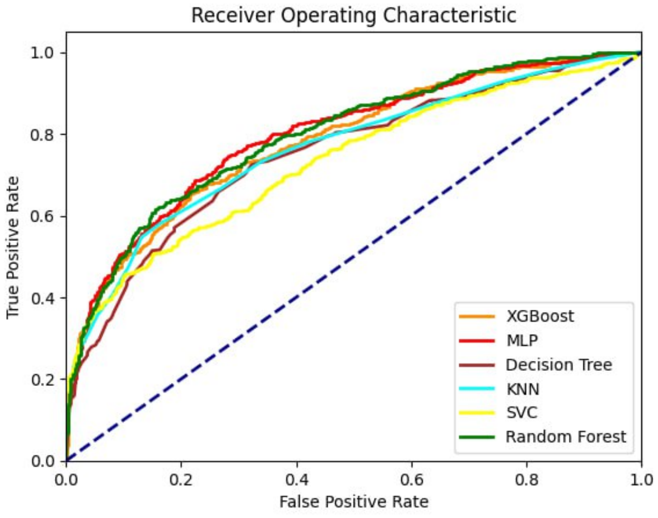

Fig. 1. ROC Curve

score between 65% and 69%, whether feature engineering was applied or not. The feature engineering introduced additional complexity and slightly improved the model's performance.

F. XGBoost

Lastly, the XGBoost classifier achieved strong performance as well. Without feature engineering, The model achieved an accuracy of 73.26% and an F1 score of 0.6472. The classification report indicated a precision of 73% for both classes, with recall values of 85% and 58%, respectively. The confusion matrix showed 638 true negatives and 321 true positives correctly identified, and the ROC-AUC score was 0.7856. Later with feature engiineering, the model achieved an accuracy of 71.43% and an F1 score of 0.6160. The classification report indicated a precision of 71% for both classes, with recall values of 84% and 54%, respectively. The confusion matrix showed 635 true negatives and 300 true positives correctly identified, and the ROC-AUC score was 0.7729.

The decrease in performance with feature engineering suggests that the additional features may have introduced noise or irrelevant information that hindered the model's effectiveness.

Overall, the results indicate varying levels of impact from feature engineering across different models. While models like KNN and MLP showed notable improvements, others like Random Forest and SVC maintained stable performance, and XGBoost experienced a slight decline.

VI. Limitations

In this study, we utilized a dataset consisting of approximately 6,500 rows, which posed a significant limitation on the generalizability of our results. Despite our efforts to locate a better dataset, this was the most comprehensive one available. Other datasets we found contained even more limited data, further constraining our research.

We attempted to enhance model complexity through extensive feature engineering, but this led to models that overfitted the training data. This overfitting was evident despite the high performance on the training sets. To illustrate, we conducted an example using Support Vector Classification (SVC) with feature engineering. The learning curve from this example showed that the training score began very high, close to 1.0, and decreased slightly as the training size increased. This pattern indicates initial overfitting of the smaller training set, but the model maintained high performance with more data. Conversely, the cross-validation score started lower but improved significantly with increased training size, approaching the training score as more data was added.

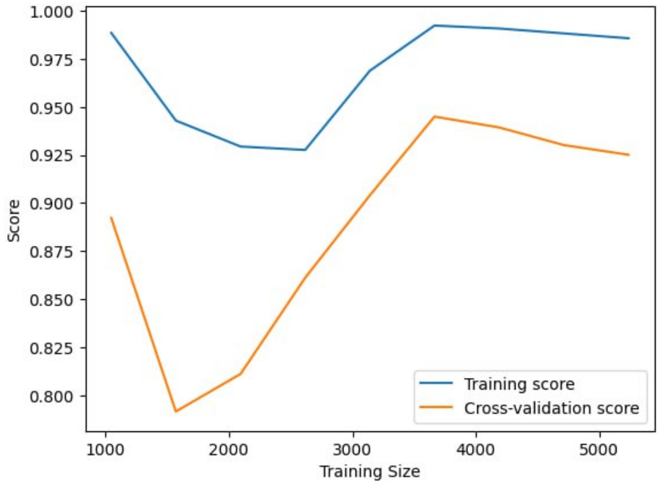

Fig. 2. Learning Curve

Analyzing these curves, we observed that the training score remained consistently high, reflecting strong performance on the training data. The cross-validation score's notable improvement with more data indicated better generalization and reduced overfitting. The narrowing gap between the training and cross-validation scores suggested that the model's performance on unseen data became more comparable to its performance on training data as the dataset size increased.

Additionally, while feature engineering was intended to enrich the dataset and improve model performance, it sometimes introduced noise and irrelevant features, particularly evident in models such as XGBoost. This highlights the challenge of feature selection and the need for careful validation of engineered features.

We explored the impact of different training-test splits on the performance of our models. Specifically, we experimented with several data splits, including 80-20, 60-40, and 50-50 ratios. These experiments aimed to determine whether the proportion of data allocated for training versus testing would significantly influence the model's performance. Notably, the Random Forest algorithm was included in these evaluations.

Interestingly, the results from these different splits showed minimal variation. The model's performance metrics, such as accuracy, precision, recall, and F1 score, remained consistent across all tested splits. Given the negligible differences in performance across these splits, we concluded that varying the training-test split ratios did not provide any substantial benefit for our analysis. This consistency in model performance across different data splits further suggests that the current dataset, despite its limitations, provided a stable foundation for evaluating model performance. However, the results did not show significant variations, indicating that the models might be hitting a performance ceiling due to the dataset's inherent characteristics.

In summary, while our dataset posed certain limitations, including its size and the challenges of feature engineering, the rigorous methodologies employed in this study ensured a thorough evaluation of model performance. The stability of results across different data splits reinforces the validity of our findings, although it also points to inherent constraints within the dataset that limit further performance improvements.

VII. Conclusion

The comparative analysis of various machine learning models revealed that the Random Forest and Neural Network (MLP) classifiers consistently demonstrated the highest performance in predicting movie success. The Random Forest model achieved the highest accuracy and F1 scores both without and with feature engineering, showcasing its robustness and reliability. The Neural Network also showed strong performance, particularly with feature engineering, indicating its capability to model complex non-linear relationships within the data.

The XGBoost classifier also performed well, although the feature engineering introduced additional complexity that resulted in a slight decrease in performance metrics. This suggests that while XGBoost is powerful, careful selection and validation of features are crucial to avoid introducing noise. The Decision Tree and KNN classifiers showed moderate performance, with slight improvements observed after feature engineering. This indicates that these models benefit from enriched feature sets but may also require more sophisticated techniques to enhance their predictive power further.

The Linear SVC, while demonstrating reasonable accuracy, had lower F1 scores compared to the other models, suggesting potential limitations in capturing the full complexity of the data. This underperformance highlights the need for linear models to potentially incorporate non-linear transformations of features or to be complemented by more complex models in ensemble approaches.

Overall, the results underscore the importance of hyperparameter tuning and feature engineering in optimizing model performance. While feature engineering generally introduced additional complexity, its impact on performance varied across models, highlighting the need for model-specific strategies in data preprocessing and feature creation. The study also revealed that certain models, such as Random Forest and MLP, are inherently more robust to the variations in the dataset, which is a critical consideration for practical applications.

Future work should focus on several key areas to build upon these findings. Expanding the dataset size and diversity would provide a more comprehensive evaluation of model performance and generalizability. Additionally, exploring advanced feature engineering techniques, such as automated feature selection, deep learning-based feature extraction, and domain-specific feature creation, could further enhance model accuracy and robustness.

In conclusion, while the current study provides valuable insights into the comparative performance of different machine learning models, addressing the identified limitations through future research will be crucial. By incorporating more sophisticated data preprocessing, feature engineering, and validation strategies, we can achieve more robust and generalizable machine learning models for predicting movie success and potentially other domains.

References

[1] J. Ahmad, P. Duraisamy, A. Yousef, and B. Buckles, "Movie success prediction using data mining," in *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–4, 2017.

[2] X. Yu, Y. Liu, X. Huang, and A. An, "Mining online reviews for predicting sales performance: A case study in the movie domain," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 4, pp. 720–734, 2012.

[3] O. D.A., S. O.M., F. A.K., A. Omotunde, O. A., A. Oduroye, W. Ajayi, and M. Yaw, "Movie success prediction using data mining," *British Journal of Computer, Networking and Information Technology*, vol. 4, pp. 22–30, 09 2021.

[4] M. Galvão and R. Henriques, "Forecasting movie box office profitability," *Journal of Information Systems Engineering & Management*, vol. 3, no. 3, pp. 1–9, 2018.

[5] T. Sharma, R. Dichwalkar, S. Milkhe, and K. Gawande, "Movie buzz-movie success prediction system using machine learning model," in *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, pp. 111–118, IEEE, 2020.

[6] S. Joshi, E. Abdelfattah, and R. Osgood, "Classification of movie success: A comparison of two movie datasets," in *2022 IEEE World AI IoT Congress (AIIoT)*, pp. 654–658, IEEE, 2022.

[7] R. Sharda and D. Delen, "Predicting box-office success of motion pictures with neural networks," *Expert Systems with Applications*, vol. 30, no. 2, pp. 243–254, 2006.

[8] K. I. Asad, T. Ahmed, and M. S. Rahman, "Movie popularity classification based on inherent movie attributes using c4. 5, part and correlation coefficient," in *2012 International Conference on Informatics, Electronics & Vision (ICIEV)*, pp. 747–752, IEEE, 2012.

[9] J. Du, H. Xu, and X. Huang, "Box office prediction based on microblog," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1680–1689, 2014.

[10] S. M. R. Abidi, Y. Xu, J. Ni, X. Wang, and W. Zhang, "Popularity prediction of movies: from statistical modeling to machine learning techniques," *Multimedia Tools and Applications*, vol. 79, pp. 35583–35617, 2020.