PREDICTIVE MODELING OF MOVIE PERFORMANCE USING ATTRIBUTES RELATED TO PRE-RELEASED MOVIES

by

THINEESHA MADHUSANKA KAHAVIDHANA

GENERAL SIR JOHN KOTELAWALA DEFENCE UNIVERSITY

2025

PREDICTIVE MODELING OF MOVIE PERFORMANCE USING ATTRIBUTES RELATED TO PRE-RELEASED MOVIES

A thesis by

T M KAHAVIDHANA

D/DBA/22/0017

Supervised by

DR. D U VIDANAGAMA

submitted in partial fulfillment of the requirement
of the award of the degree of

BACHELOR OF SCIENCE HONOURS IN DATA SCIENCE AND BUSINESS ANALYTICS

Department of Computational Mathematics
Faculty of Computing
General Sir John Kotelawala Defence University
Sri Lanka 2025

DECLARATION

I declare that this is my own work and this thesis does not incorporate without acknowledgment any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. Also, I hereby grant KDU the non-exclusive right to reproduce and distribute my thesis, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

………………………………………. Date:……12/22/2025….……...

T M Kahavidhana
Departement of Computational Mathematics
Faculty of Computing
General Sir John Kotelawala Defence University
Sri Lanka

The above candidate has caried out research for the partial fulfillment of the BSc Honours in Data Science and Business Analytics Degree thesis under my supervision.

………………………………………… Date:…23/12/2025……………………

Dr. DU Vidanagama
Head of the Department
Senior Lecturer
Departement of Computational Mathematics
Faculty of Computing
General Sir John Kotelawala Defence University
Sri Lanka

ABSTRACT

The global film industry is a massive industry with huge financial risks due to the unpredictability nature of measuring its box office success and audience reception. In this research, it provides a method to minimize this unpredictability by using a robust classification model that based on machine learning, that classifies movies into success or unsuccess by using user preferences based on IMDB ratings. The method provide by this research uses pure pre-release data that gathered from IMDB and TMDB with narrative information from Wikipedia for 1400 US films between 2000 to 2023. This study includes time sensitive historical movie performance measure for actors and directors to avoid data leakage. Also, those meta data of a movie is integrated with plot summaries that have been transformed into dense vector embeddings by using a longformer model. The optimum model developed in this research combines with stacking ensemble XGBoost and LightGBM base learners with Logistic Regression meta-learner.

The developed model achieved accuracy of 77% and F1-score of 77% on an unseen data set. Feature important analysis reveals that duration of a film and the director with successful previous movies and semantic features extracted from plot synopsis can be considered as the best predictors to classify the IMDB rating category. This study addresses impactful methodological gaps in the existing literature by being able to combine meta data and deep narrative data to ensure rigorous temporal validity. The final model deployed as an interactive web application with content based suggestions system that provides robust suggestionstions that can apply in real world scenarios, presenting stakeholders with an evidence-based framework to reduce its financial and maximize its investment strategy during the key-production phase.

Keywords: Movie Success Prediction, Machine Learning, Pre-Release Forecasting, Natural Language Processing (NLP), Ensemble Methods

DEDICATION

With all my heart, I dedicate this work to the pillars of my support. To my parents, whose love and encouragement provided a great source of motivation and love, and to my supervisor Dr. D. U. Vidanagama, who guided and supported me intellectually throughout this journey and every obstacle I encountered. And to all my lecturers, who provided me an great inspiration and equipped me with knowledge to undertake this research and also in my university life. Thank you for believing in me.

ACKNOWLEDGEMENT

With all my heart, I would like to express my heartfelt gratitude every individual who helped and inspired me through this journey. Without them this would not be a possible to achieve.

My heartfelt gratitude goes to my project's supervisor Dr. DU Vidanagama who guides me throughout this research project and help me in in every situation every obstacle that I face during this research project. When I lost with some complexities in this research project, she guides me and give patience and give encouragement with her intellect.

Also, I would like to thank General Sir John Kotelawala Defence University for giving me space and resources to do this research and also to the Faculty of Computing and the dean, Head of the department of Computational Mathematics and all the lecturers, instructors and all other staff. You helped to build my academic journey into a success story.

I also like express my heartfelt thanks to LOLC Finance Consumer and Digital Business Unit Head and Chief Digital Officer Mr. Hasala Thilakaratne and Head of Credit Cards Mr. Yasantha Samaranayaka and my supervisor Mr. Joyson Yogaraj to give me the freedom that I need to do this research even in the official environment.

I would also like to thanks to my fellow classmates and my friends of me to be always there with me in every obstacle that I face during my academic journey and helped me.

This study heavily uses the publicly available data and also the open-source libraries and free API keys. I would like to express my appreciation for IMDB, TMDB and Wikipedia for providing me all the required resources and data that need to complete this research.

Finally, I would like to thank my parents my brothers and all of my friends to believe in me and be there with me.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBERVIATION

IMDB	Internet Movie Database
TMDB	The Movie Database
XGBOOST	Xtreme Gradient Boosting

CHAPTER 1
INTRODUCTION

1.1 Overview

The global film industry is an industry with mixture of art, culture, and business with artistic creativity and intersects with multi-million-dollar investments. Studios, stream platforms, and independent filmmakers invest huge amount of their money into make film and bring it onto a big screen. Those investments can be varied, they can be low budget independent production movies or they can be high budget franchise blockbuster movies with multi-million-dollar investments. But for all these movies there is one kind of similarity, that is its unpredictable nature due to its complexity. Commercial and critical success of a film is unpredictable by its nature and even high budget films can be flop in box office and also can get negative reviews from critiques also while low budget film can surprise us all by getting huge commercial success and positive critical responses.

The global film industry is one of the impressive and remarkable industry with great creativity and but inherent with its unpredictable nature. The global film industry generates multi billion dollars annually with the combination of forces like overseas markets, streamers and theatrical releases. The United States is the largest contributor for this global box office income with its large annual revenues, in 2025 alone US reaches into over 8 billion profits. This much of a revenue, highlights that despite of the evolving marketing trends and differentiate consumer preferences and shifting marketing trends, film industry holds a special place of a country's economic growth. This much of an financial scale gives the point that movie release is not only about the its financial investment but also its countless creative effort that film makers give [1].

While movie industry being one of the most interesting industry, box office factor gets directly or indirectly influenced by various factors like story quality, cast strength, directional reputation, production scale, movie duration, release time, and market competition. Among these, perception driven indicators like reviews, audience ratings, and word-of-mouth plays a huge role in gaining public attention for the film and commercial success. For this statement there are some empirical evidences also like, one study finds out that Bollywood films have strong correlation between critic and audience ratings and box office revenue after controlling for production budget, screens, and distribution. Such findings give the suggestions like audience perceptions and pre-release movie bus can have some impact on movie's commercial success [2].

Apart from these aggregated ratings, source of that rating comes from also matters. Empirical evidences provide that; there is a difference between professional reviewers and normal consumers impact on the film and their impact for the movies differs. It founds out that average consumers are playing a huge role when it comes to overall variation of a movie performance in terms of outcomes. While professional critics are more stable with their judgement and also, they are very consistent. Also, it founds out that genre familiarity also a key factor, more familiar genres get more influence by normal consumers while non familiar genres get more influence by professional critics. This study shows that the different of normal consumers and professional critics influence on the film [3].

The market and financial uncertainties give significant strategic and financial challenges for film producers, studios and streaming platforms. The decision-making process for its marketing, financial, and release strategies are more commonly made even before the films debut to the screen months or years before and those decisions are heavily influenced by factors like overall historical experience of filmmakers, the gut feeling, and some reputation features. Because of this whole process is based huge human involvement in decision making there can be huge financial risks and tougher competition in both theatrical and digital markets. However, because of these factors in recent past the industry becomes more data driven because with the availability of vast pre-release information like meta data of a film and plot details.

The rapid growth of machine learning technique and predictive analysis opens the doors to reduce uncertainty in this industry because by using pre-release data these techniques can provide detail-oriented evidences by methodologically analysing the pre-release features ensuring the evidence-based critical and box-office forecasts so that stakeholders can make better data driven decisions. These machine learning models have the capability of discovering the complex patterns that humans can't by combining quantifiable production metrics with qualitative narrative data to construct more comprehensive predictive frameworks.

This research is a combination of both storytelling and data science. This research aims to develop methodologically rigorous and temporally precise predictive framework for forecasting film success in terms consumer perception based on IMDB ratings before film's theatrical release. This research integrates the quantitative meta data analysis with advance narrative representation techniques. The main focus of this study is to transform the early-stage film decision making, means before stating the production of a film, in addition to contributing to the academic field by predictive modelling, it also provides practical value for industry professionals by reducing investment risks, improving resource allocation, and supporting more strategic choices in today's highly competitive movie market.

1.2 Background

Traditional decision making about a film advance from concept to production with heavily relied on creativity, reputation, experiences, and instinct feelings. Studio producers, investors, independent film makers, and executives often make their decisions by looking at the simple factors like the popularity of a cast and crew, the story strengthens according to their previous experiences, and prevailing market trends. This instinct based decision can make a blockbuster hit like previous some of the blockbuster hits but also there is a high risk because it can be unsuccessful also, because there are some scenarios that high budget, well product films get flopped in box-office wit negative consumer responses. Also, it can be in other way around also, modest independent films can achieve unexpected success also.

This industry is particularly unpredictable with huge amount of stake and investment and if it is failed, it will have to pay big consequences. In 2023. the global box office generate approximately $33.9 billion, with over $9 billion coming from the North America alone[1]. This finding highlights the importance of this industry in terms of economic and also the high stakes involved in decision making. Because of this high unpredictability, single misinformed or misguided information can lost multi-millions because of that making accurate forecasting more crucial than ever.

Since the development of data science and machine learning in the last few years, new ways have appeared for optimizing pre-release prediction. The existence of vast quantities of structured data (e.g., budget, genre, casting information) and unstructured data (e.g., plot description or keywords) has provided the means for building predictive models which foretell the probable performance of a movie before release. Most existing models, however, have serious limitations. Some inadvertently use post-release data, resulting in leakage of data, while others fail to account for changing reputations over time. Also, narrative content and production data are usually partitioned and examined, diminishing their overall predictive power.

To respond to such gaps, scientists are now looking more for hybrid modelling approaches that combine multiple pre-release data sources methodologically. A temporally sound and stable predictive model can provide stakeholders in the film industry valuable insights ahead of investment, and thereby reduce uncertainty and improve decision-making. This study is motivated by that intent to enable more data-enabled and informed strategies for the film industry.

1.3 Motivation and Relevance of the Study

The film industry is a high-investment and correspondingly high-uncertainty setting. Despite the huge and significant amounts of money and time invested in production and marketing, the financial success and critical success of films is hard to predict. Traditionally, investments, film planning and marketing campaigns has been made on the basis of individual judgment, experience, and past performance. While this tactic has worked before, it also poses enormous financial risk to producers, distributors, and investors. With more and better data, and better analytical methods, there is a need for more objective, evidence-based decision-making tools in the movie industry.

Accurate pre-release prediction can allow decision-makers to determine the viability of a movie before it is released onto the marketplace. It is an approach that can allow strategic planning with regard to distribution, marketing expenditure, and production level. Most current forecasting techniques are derived from post-release information like viewers' ratings or box-office revenues and are less useful in actual-world decision-making. This paper aims to bridge the gap between data-driven forecasting and creative choice. By integrating structured pre-release information such as budget, cast, genre, and crew with the unstructured information of plot summaries, we can construct a model of prediction that functions under realistic pre-release circumstances. The system can offer probabilistic performance predictions with a certain level of expectation, minimizing uncertainty and allowing for better resource management. This study is significant as it helps develop an up-to-date and composite predictive model for film performance classification. This model is different from previous ones in that it only uses pre-release information to guarantee that predictions are practical in real production environments. More informed decisions are made by the producers, marketers, and investors through the results of this study, resulting in productive investment decisions and higher rates of success in the film business.

1.4 Research Problem

The global film industry is where art meets commerce, with billions of dollars spent each year on films whose commercial success is by no means guaranteed. Even with greater access to historical data, the views of experts, and high-level analysis methodologies, it remains an elusive task to accurately forecast the success of a film before it is released.

Uncertainty disproportionately affects fiscal matters. Flop movies can cost a lot of money the studios and production houses lose, and change studios' and production houses' course. In such cases, evidence-based greenlighting factors are crucial to reduce financial risk and optimize investment strategy. In recent years, data-driven methodologies have been an effective predictor of performance in creative industries. The effects of star power, budget, genre, and reception on box office performance have been studied. For example, evidence from research shows a significant positive correlation between audience and critic ratings and box office returns, implying that early anticipation can determine commercial performance.

Similarly, comparisons similarly indicate that professional critics and general audience affect early expectations, with consumer ratings playing a larger role in known genres and critic ratings in unknown ones. These proofs attest that early predictors and content-based variables must be integrated into prediction models. But even with all these innovations, most of the available predictive models are, in fact, flawed. Most models misuse post-release data, for example, user ratings or box office results, that causes data leakage and undermines the accuracy of predictions in actual pre-release scenarios. Other models use static measures of talent reputation that do not account for how a contributor's career develops over time. Also, narrative data (e.g., plot structure, themes) and metadata (e.g., cast, budget, production) are usually studied in isolation and not as parts of an integrated prediction system.

This disjointed approach limits the accuracy, efficacy, and usability of current models. Therefore, there still remains a gigantic gap between academic needs and industry requirements. Though current literature lays out the capabilities of data-driven prediction, the absence of an integrated, time-bound, pre-release predictive system prevents its efficient use for high-risk decision-making. A competent platform that brings together standardized metadata and unstructured narrative analysis, without the addition of post-release information, can be able to revolutionize how risk is determined and resource allocations are made by producers, distributors, and investors. Closing this research gap will help enhance academic comprehension as well as real-world application of predictive modeling within the film industry.

1.5 Research Aim

The aim of this project is to build a machine learning–based system that can estimate a film's IMDb rating category before it is released by analysing pre-release information such as film metadata, the reputation of contributors, and plot narratives. In addition, the study seeks to offer practical, content-based suggestions that help filmmakers and investors make better strategic decisions and minimize financial risk in the film industry.

1.6 Research Objectives

Research Problem While data-driven approaches have been extensively used in the film industry, it has not been possible yet to forecast a film's performance category prior to release using a valid, up-to-date, and uniform method. All current models are plagued by methodological shortcomings, for example, applying post-release data (resulting in data leakage), delayed talent measures, or isolating metadata from narrative content. This deficiency undermines the practical usability of forecast models for pre-release decision-making.

In order to solve this, the current study intends to create, implement, and test a machine learning prediction model using structured metadata, contributor reputation, and plot sentiment to forecast IMDb rating categories before release. The particular objectives are to:

1. To identify significant features between metadata, star reputation, and plot sentiment to learn about their predictiveness.

2. To create new features to improve the model's prediction accuracy while maintaining temporal relevance.

3. To train and evaluate machine learning models for IMDb rating classification with robust validation.

4. To deploy the best-performing model to supply real-time prediction data and content-based suggestions through a web-based platform.

1.7 Thesis Structure

This thesis consists of six chapters, which contribute to building a robust, transparent, and viable prediction model for the prediction of film performance:

- Chapter 1: Introduction
  This chapter gives the background, motivation, research problem, aims, and general study structure.

- Chapter 2: Literature Review
  This chapter compiles previous research on movie prediction with a focus on methodological enhancement, limitations, and key research gaps.

- Chapter 3: Methodology
  This chapter outlines the overall methodology with research design, data sources, feature engineering strategy, modeling techniques, and evaluation plan used in this study.

- Chapter 4: Results and Evaluation

  This chapter outlines the analytical process, model training, performance measurement, and system deployment for the proposed framework.

- Chapter 5: Conclusion and Future Works

  This chapter summarizes the major contributions of the work, offers key takeaways, and makes educated guesses about possible directions for further work.

CHAPTER 2
LITERATURE REVIEW

2.1 Overview

The global film industry is a risky commercial enterprise in which enormous financial outlays must be committed under great uncertainty about how the market will react. Billions of dollars are invested annually in the production of films, from low-budget independent productions to enormous blockbusters costing hundreds of millions of dollars. Box office and critical uncertainty remain, however, a dominant concern among producers, investors, and distributors along the industry value chain. With so much money on the line and so many variables influencing the commercial success of a film, including story quality, star power, marketing quality, competitive release date, and consumer demand, achieving the commercial success of a movie prior to its release has become an intellectual challenge with strategic uses as a risk mitigation strategy and sound decision-making tool for the entertainment industry.

Contemporary film success forecasting has been enabled by developments in machine learning and data science, and research has utilized a broad variety of methodological approaches and sources of information. Researchers have leveraged structured metadata in film databases, unstructured text data in plot summaries, real-time social media streams, web search volumes, and sophisticated ensemble and deep neural network models. This development of analytic techniques is keeping up with increased consciousness of forecast quality as a strategic competitive resource and availability of computing capacity for handling heterogeneous, high-dimensional data from varied sources.

However, for concerted reading of phenomenal fiction, there are unavoidable methodological errors that limit the application of most forecasting models in the real world to genuine pre-release prediction. One of the very significant concerns is the utilization of post-release information such as user rating counts, overall box office returns, and viewers' reviews as predictor variables in models when models are explicitly being designed to predict pre-release. Although application of such result measures artificially inflates reported model performance on look-back tests, it renders such models meaningless if intended function is to provide decision-making recommendations to investors who must make most crucial investment decisions during pre-production phases. This overview solely criticizes predictive method development, reaches endemic problems with current methods, and positions itself for addressing significant research deficits in actually pre-release forecasting systems.

2.2 Evolution of Metadata-Based Prediction Approaches

2.2.1 Early Models and Data Limitations

Initial attempts at prediction of film success were based on structured metadata, such as genre, director, cast, writers, country, runtime, and budget. These initial studies employed simple machine learning techniques like simple linear regression, multiple linear regression, and simple neural networks to attempt to create baseline prediction capability [4]. Though these early models demonstrated that it was feasible to use computation to forecast box office success, they were troubled by a fundamental methodological flaw that has persisted throughout much of the subsequent literature.

One mistake made in some of the initial studies were the inclusion of post-release data in prediction models. The majority of research employed variables such as user vote counts, total gross box office revenue, total box office, and user ratings as predictor variables[5], [6], [7], [8]. For instance, variables such as "num_voted_users" or "total_box_office_revenue" are outputs and not predictors, effectively rendering the models useless for actual pre-release prediction. This data leakage issue with the data meant that while these models were extremely well-performing in backtesting, they couldn't provide stakeholders with useful insights during the important pre-production phase of investment.

2.2.2 Methodological Progress in Pre-Release Prediction

As the awareness about issues of data leakage grew in research community, authors began developing more methodologically improved and rigorous approaches with strict pre-release metadata usage. This came alongside the application of more advanced machine learning algorithms, particularly ensemble techniques such as Random Forest, XGBoost, and

Gradient Boosting. These newer methods displayed amazing gains in performance upon deployment in carefully curated pre-release feature sets.

Large-scale experiments on well pre-processed IMDb datasets achieved excellent results: 92.7% accuracy was achieved by Random Forest models [5], 90.39% by XGBoost [16], and 83% accuracy by Gradient Boosting [4]. The results were particularly impressive as they were achieved solely from pre-release data, showing that strong predictive signal is embedded within metadata available at the decision point. The performance of the above models illustrated the necessity of rigorous data preprocessing, including complex missing value imputation, resilient categorical encoding (one-hot encoding), and extensive outlier scrutiny [7].

2.2.3 Metadata Sources and Datasets

Most metadata-based studies leverage multiple online sources to construct large datasets. Internet Movie Database (IMDb) is the primary metadata source for movies that provides features like cast, crew, directors, writers, genres, runtime, budget, ratings, and vote totals [4], [5], [7], [8], [9], [10]. The Movie Database (TMDb) supplements IMDb information with additional data about production, including production companies, countries, and release dates [11], [12]. Box Office Mojo provides financial data in the form of domestic and foreign gross revenues, which are essential in assessing levels of success [9].

Data Collection process is different across varies studies. The data collection process for some researches did the manual curation of datasets by choosing films released within specified time periods (e.g., 1980–2020 [7] or 2002–2012 [4]) to ensure temporal relevance and eliminate outdated trends is one process. Others employ web scraping scripts and APIs to scrape data automatically and merge data from different sources by common identifiers in data sources such as movie names and release years [11]. Dataset sizes range from a minimum of 400 movies [13] to over 32,000 movies [4], with the bigger dataset sizes typically resulting in stronger models.

2.2.4 Advanced Features from Historical Performance

A significant advance in metadata-based prediction was coming through with the development of features that represent the historical performance of key film contributors. Rather than looking at actors, directors, and other personnel key contributors as simple categorical variables, scholars began quantifying their record employing reputation-based measures [8], [10], [11], [14]. This method transforms static categorical handles into informative numerical representations carrying ideas such as "star power" and director reputation [15].

Earlier approaches in the literature were to apply static historical averages, whereby a single average IMDb rating was used for an actor or director and then uniformly applied to all their films independent of the forecast horizon [4], [8]. These static historical values do not reflect temporal dynamics or variations over time in individual performance. This deficiency creates a divergence between real world career development and the predictive modeling process. Thus, the models used in these analyses may not be as good at explaining how an actor or director's reputation, level of skill, or audience acceptance can change throughout his or her career, potentially constraining the accuracy and adaptability of their estimates.

Deep learning algorithms, e.g., 1D Convolutional Neural Networks (CNNs), have been particularly good at leveraging these historical feature vectors [11], [14] . These algorithms are capable of discovering high-level patterns and interactions among historical performance data and making more objective and precise prediction than the traditional approach, which relies on post-release social signals.

2.2.5 Feature Importance and Model Interpretability

Studies that examining the feature importance of authors, actors, and directors founds out that these features significantly affect ratings and box office performance [4]. Authors develop core story elements like plot, characters, and structure, while actors give star power which attracts audiences. Production budget and MPAA ratings (content age ratings) also significantly impact revenue as well as reach [6], [8]. Conversely, factors such as runtime and shooting locations have weak direct correlations with success but affect viewer decisions indirectly [4].

Decision tree-based models (Random Forest, XGBoost) provide intrinsic interpretability in the form of feature importance scores, and thus they are beneficial to stakeholders who desire actionable results [5], [6], [8], [9]. Neural networks, although performing better in terms of accuracy, lack good interpretability in the form of intricate interconnecting weights and hidden layers [4]

2.3 The Emergence of Social Media and Web Predictors

2.3.1 Social Media Interaction as Predictor Signal

Parallel to the development of metadata-based approaches, researchers began to research the forecasting value of real-time social media usage data. This strand capitalized on the growing influence of digital marketing and social media platforms in shaping the anticipation and reception of viewers. Experiments employed diverse social signals like YouTube trailer metrics (YouTube views, likes, dislikes, comments) [16], Twitter metrics (range of tweets, hashtag usages) [12], [16], Facebook engagement (actors' page likes, directors' page likes, film page likes) [17], and Wikipedia pageview statistics [12].

High-level regression models exhibited substantial correlations of social media metrics and IMDb ratings, with correlation coefficients as high as 0.8915 in some of the research [16]. The most predictive features were often like-to-dislike ratios of the trailers of movies and sentiment analysis of social media messages [16]. Nonetheless, research consistently demonstrated that single social media features were insufficient for prediction accuracy and needed to be combined with traditional metadata to function optimally [12], [17].

2.3.2 Social Media Model Datasets and Feature Sets

Social media research utilizes APIs of websites such as Twitter, YouTube, Facebook, and Wikipedia to mine engagement metrics. In the case of Twitter, researchers collect tweets with movie names within constricted time windows and extract features like tweet counts, occurrences of hashtags, and mentions [12], [16]. Sentiment analysis tools are utilized to spot positive or negative tweets and generate polarity scores and subjectivity scores [12].

YouTube metrics include views, likes, dislikes, comments, and favourites for authentic movie trailers [16]. The like-to-dislike ratio is an especially strong predictor [16]. Wikipedia provides pageview counts for movie-related pages as a proxy indicator of public interest [12]. Facebook metrics include page likes for movies, directors, and actors, although studies show that Facebook metrics alone are not accurate enough and must be cross-checked with metadata for predictive value [17].

2.3.3 Temporal Limitations of Social Media Approaches

Though their forecasting ability, social media-based approaches have built-in time limitations that render them unsuitable for real pre-release forecasting. These approaches primarily derive marketing effectiveness and short-term viewership interest prompted by promotional campaigns, which typically happen weeks or months prior to release. Though appropriate for near-release forecasting and advertising optimization, these approaches are not capable of producing early-stage information needed for fundamental production and investment decisions.

Web search pattern analysis based on Google Trends reports presented a new method for quantifying public interest trends [13]. Time-series search frequency of movie titles and actors transformed to static feature vectors were analysed, and high classification accuracy of 72.25% was achieved using Support Vector Machine models [13]. Interestingly, it was observed through research that persistent, long-term search interest was more related to the quality of the movie than short-term hype-driven activity, which provided some insight into the distinction between marketing-hyped activity and genuine audience expectation [13].

2.4 Forecasting Using Plot Content

2.4.1 Narrative Analysis: Sentiment and Sequence

The most ambitious advancements in movie prediction research has been the development of models that analyze narrative content directly from plot summaries and scripts. This approach is the earliest available timing of prediction, enabling examination of a story's capability prior to significant financial outlay. Early efforts at this were attempts at using sentiment analysis techniques in trying to quantify the emotional trajectory of films.

Researchers developed techniques like Sentiment Arc using techniques such as VADER sentiment analyzer to plot summaries sentence by sentence, developing vectors that captured the sequence of positive and negative emotional events throughout the story [18]. These sentiment vectors were then processed by deep learning models, namely Long Short-Term Memory (LSTM) networks and 1D CNNs, which are particularly well-suited to represent sequential data and capture long-range dependencies in narrative emotional arcs [18].

2.4.2 State-of-the-Art Semantic Models

The breakthrough in ploy-based forecasting came with the introduction of contextual word embeddings, namely ELMo (Embeddings from Language Models) [18] and BERT (Bidirectional Encoder Representations from Transformers) [10]. Both of these state-of-the-art NLP models enable the encoding of rich semantic and contextual information from plot descriptions beyond mere sentiment analysis to capture subtle narrative elements.

Better methods utilized additional structural features derived from character descriptions, including Subject-Verb-Object (SVO) triplets that recognize fundamental narrative relations [10]. BERT models achieved high classification accuracy rates of more than 73% in the prediction of popularity of thriller movies and 70% prediction accuracy for quality in the action genres [10]. Notably, such models performed particularly well at identifying likely failures—a task that could prove particularly valuable for risk-averse investors seeking to "avoid losers" as much as merely selecting winners [18].

2.4.3 Data of Narrative-Based Models

The CMU Movie Summary Corpus, with plot summaries, character descriptions, and genre information for thousands of movies [18], is used by most narrative-based studies. Supplementary data from IMDb and TMDb are scraped to provide plot synopses and overviews to supplement the corpus [10]. Rotten Tomatoes ratings (Tomatometer and audience rating) as target variables for predicting success are utilized [10], [18].

Preprocessing consists of tokenization, lemmatization, removal of stopwords, and stemming to preprocess text data. BoW representations, TF-IDF vectorization, and topic modeling (Latent Dirichlet Allocation) are employed to extract topical features from plot texts [14]. More advanced research utilizes pre-trained embeddings such as GloVe, ELMo, and BERT to generate dense vector representation that captures semantic meaning [10], [18].

2.5 Algorithmic Evolution in Modeling

2.5.1 Traditional Machine Learning Foundations

The algorithmic evolution of film prediction studies also follows a clear path from simple statistical models to complex machine learning and deep learning frameworks. Early studies relied predominantly on extensions of linear regression and simple neural networks, which although providing rudimentary insights could not model complex feature interactions efficiently [4], [16], [17]. As datasets grew and became richer in information, researchers sought more powerful machine learning techniques.

Classic machine learning methods like Decision Trees, Random Forest, Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), and gradient boosting methods became the standard tools for movie prediction applications. The methods proved particularly effective when coupled with considered feature engineering and preprocessing. Experiments on ensemble methods always outperformed the single-algorithm approach, with XGBoost and Random Forest turning out to be particularly robust choices for classification as well as regression tasks [5], [7], [8].

Specific instances are bagging (Bootstrap Aggregating) over Random Forest and XGBoost for even improved prediction using variance reduction and avoidance of overfitting [7]. Naive Bayes classifiers are employed for text-based sentiment classification due to their simplicity and efficiency with high-dimensional data [9]. k-NN and SVM models are applied for regression and classification, and kernel functions enable non-linear decision boundaries [15], [17].

2.5.2 Deep Learning and Complex Architectures

Deep learning was a significant step in the discipline's capacity to handle high-dimensional, sophisticated data as well as detect faint feature interplay. Convolutional Neural Networks (CNNs) were especially helpful in handling narrative sequential information as well as structured metadata. Generative CNN models were capable of capturing correlation among heterogeneous movie features such as genres, budget, cast structures, as well as plot descriptions simultaneously [14].

Tabular data is suited best by 1D CNNs and achieves significant performance boost relative to baseline models [11]. LSTM networks perform well to learn temporal relationships in sentiment arcs based on plot summaries [8], [18]. Hybrid models that combine BERT embeddings with CNN or LSTM layers combine textual and metadata features into multimodal prediction [10], [18].

More advanced architectures such as factorization machines have been particularly promising for handling sparse high-dimensional categorical data prevalent in movie metadata [12]. These models are best suited for finding interaction between categorical features without sacrificing computational efficiency and are hence suitable for production-level deployment. Factorization machines reduce mean squared error (MSE) to a significant extent compared to linear regression benchmarks when deployed with metadata and social media features together [12].

2.5.3 Hyperparameter Tuning and Optimization

Deep learning algorithms require accurate hyperparameter tuning to yield best performance. Research employs grid search, random search, and Bayesian optimization to find best learning rate settings, batch sizes, dropout rate, hidden layer count, and neuron counts. TensorBoard is employed for visualization and monitoring training processes, and iterative architecture tuning [8].

Regularization techniques like dropout, L1/L2 regularization, and early stopping are employed in order to prevent overfitting, especially while training on smaller datasets [11]. Techniques of cross-validation are employed for ensuring proper evaluation and generalization on unseen data [5], [7], [12].

2.6 Evaluation Metrics and Performance Benchmarks

Rigorous evaluation methods are the foundation to model verification in regression and classification problems. In regression rating prediction, Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) are usual measures [6], [11], [12], [14], [16]. Hit Ratio measures approximate the ratio of predictions within acceptable error ranges. [14].

Classification issues employ accuracy, precision, recall, F1 measure, and Cohen's kappa statistic to acknowledge model performance [5], [6], [18]. Confusion matrices and ROC plots provide diagnostic information, particularly for imbalanced success classes data sets that are underrepresented. Kappa above 0.60 indicates substantial agreement more than chance and good model performance [7].

Cross-validation techniques, particularly 10-fold cross-validation, are widely applied to give performance estimates free of bias and prevent overfitting [11]. The rank correlation coefficient of Spearm is employed to estimate the size of monotonic relationships between predicted and observed ratings [16].

2.7 Critical Research Gaps and Opportunities

2.7.1 Separated Approaches in Current Literature

A thorough investigation of current literature shows a persistent divide between various predictive paradigms. Though metadata-based paradigms [5], [8] and narrative analysis methodologies [10], [18] have both exhibited significant predictive capability, nearly no experiments have been able to combine these complementary paradigms into common hybrid frameworks. This divide is an untapped potential because the marriage of measurable production qualities with deeper narrative comprehension can potentially provide better predictive performance.

2.7.2 Temporal Inconsistencies in Historical Feature Engineering

Past approaches to historical feature engineering in movie success prediction have inherent temporal inconsistencies that undermine their chronological reliability. Most of the past works rely on static historical values, assigning the same average rating to an actor or director regardless of when the prediction is to be made [4], [8]. This overlooks the dynamic career path process, shifting audience perception, and shifting industry context, thus introducing anachronistic bias into prediction models. Despite the clear temporal character inherent in historical performance data, few studies have employed time-sensitive historical computations, where an actor's or director's historical average is derived exclusively from films acted in or directed by them before the specific film being predicted [11]. This absence of temporal consistency in the construction of historical features is a primary methodological flaw that this current study seeks to address.

2.7.3 Lack of Integration of Recommendation Systems

While accuracy in prediction has been of primary importance, integration of recommendation systems within the literature with potential to realize predictive foresights into business intelligence has lacked impetus. The gap between prediction and decision-making is an unexplored field with extremely practical business implications for business actors.

2.8 Identified Research Gap

The identified research gaps address the need for hybrid approaches that capture the strengths of multiple prediction paradigms without compromising strict pre-release data constraint adherence. The optimal system would integrate:

- Dynamic Historical Metadata: Temporally derived contributor performance metrics capturing their history leading up to the prediction moment, a core approach yet to be addressed in existing literature.

- Deep Narrative Analysis: BERT-based representations of plot abstracts that maintain semantic information and narrative structure.

- Rigorous Temporal Validation: Validation of all the features to ensure they capture information available at prediction time.

- Unified Recommendation Framework: Translation of predictive insights into business actionable recommendations.

Such a combination of hybrid approaches addresses the inherent limitations of existing studies and provides a foundation for truly practical pre-release movie success prediction that is capable of informing high-stakes industry decision-making at the earliest stages of movie development.

2.9 Summary

Film success prediction literature validates a research corpus that has evolved from the initial metadata models to sophisticated systems with social media analytics and narrative analysis. Yet, persistent methodological issues, primarily the utilization of post-release data and disengagement of complementary methodologies, limit the practical application of most existing models. The gaps in research identified suggest the need for hybrid approaches combining dynamic historical metadata with sophisticated narrative analysis under tight pre-release data restrictions. Filling the gaps is both a significant research undertaking and a path toward more useful predictive tools for the industry.

CHAPTER 3
METHODOLOGY

3.1 Overview

This chapter outlines the methodology framework built to predict film performance prior to cinema release using pre-release attributes and narrative content. The movie industry is characterized by high financial risk in the initial phases of production, where, major decisions on significant investment, in most instances tens or even hundreds of millions of dollars must be made in the absence of definite proof of downstream audiences' reception. This research can surmount that disadvantage by developing prediction models based on pre-release information, offering data-driven decision-support to producers, studios, and investors.

The primary objective of the research is to design a binary classifier that would classify the movies as "Successful" or "Unsuccessful" based solely on information available before release on the basis of IMDb rating. This method of categorizing by genre directly fulfils industry practitioners' real requirements for making binary investment decisions (greenlight or pass) rather than providing definitive numerical rating predictions. IMDb ratings provide a universally recognized, standardized measure of audience acceptance in which direct genre-to-genre, budget class-to-budget class, and release strategy-to-release strategy comparisons can be made.

To further contribute practical usefulness to practice professionals, the deployment system is equipped with a content-based suggestion module as an integrated supporting function. The suggestion module supplies historically comparable successful films as strategic models as well as forecasting output.

Several methodological innovations distinguish this work from previous and existing work:

- Strict temporal validity in that it only uses pre-release data, without contaminating data from post-release variables such as box-office performance, critic reviews, or social media sentiment.

- Time-sensitive historical performance metrics for actors and directors, which are computed dynamically from their chronologically prior work rather than static career averages.

- Structured and unstructured data integration via combination of metadata (genre, MPAA rating, runtime, budget) and narrative content (plot synopses) using transformer-based embeddings with Longformer.

- End-to-end classification pipeline with organized hyperparameter tuning, ensemble methods, and deployment architecture with integrated recommendation feature.

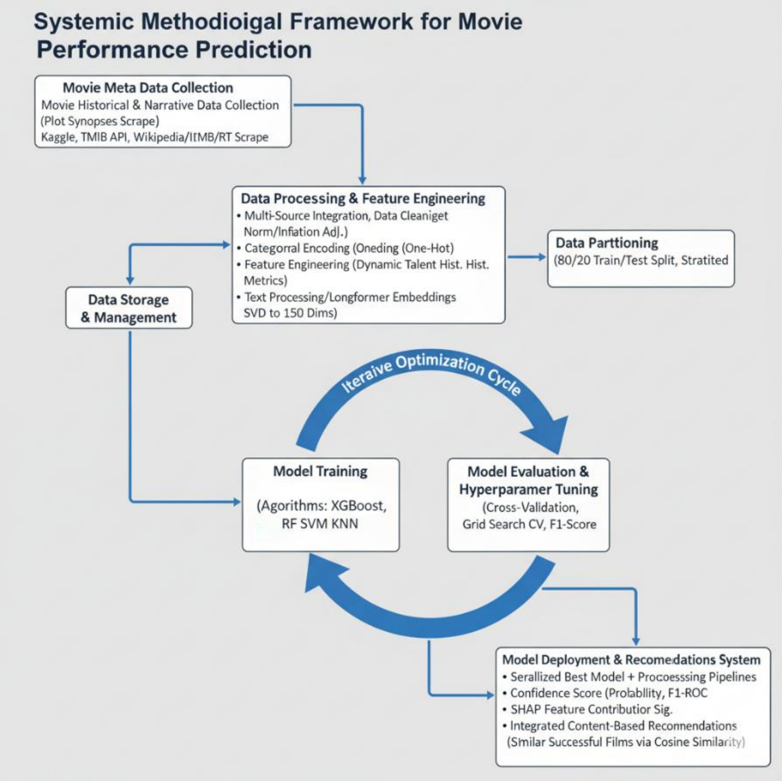

Systemic Methodioigal Framework for Movie Performance Prediction
Movie Meta Data Collection
Movie Historical & Narrative Data Collection
(Plot Synopses Scrape)
Kaggle, TMIB API, Wikipedia/IṮMB/RT Scrape
Data Processing & Feature Engineering
• Multi-Source Integration, Data Cleaniget Norm/Infiation Adj.)
• Categorral Encoding (Oneding (One-Hot)
• Feature Engineering (Dynamic Talent Hist. Hist. Metrics)
• Text Processing/Longformer Embeddings SVD to 150 Dims)
Data Parttioning
(80/20 Train/Test Split, Stratited
Data Storage & Management
Iteraive Optimization Cycle
Model Training
(Agorithms: XGBoost, RF SVM KNN
Model Evaluation & Hyperparamer Tuning
(Cross-Valldation, Grid Search CV, F1-Score
Model Deployment & Recomedations System
• Serallized Best Model + Procoesssing Pipelines
• Confidence Score (Problbllity, F1-ROC
• SHAP Feature Contributior Sig.
• Integrated Content-Based Reconnendations
(Smilar Successful Films via Cosine Similarity)

Figure 3. 1: Methodological framework for movie performance prediction

As illustrated in Figure 3.1, the systemic methodological framework operationalizes these stages through a structured data-to-deployment pipeline of metadata acquisition, preprocessing, feature engineering, model training, evaluation, and deployment under an iterative optimization cycle for continual performance improvement.

The methodology framework includes various phases to ensure scientific validity, transparency, and reproducibility:

- Research type and design specification.
- Multi-source data collection and integration.
- Extensive data preprocessing and feature engineering.
- Exploratory data analysis to inform modeling decisions.
- Predictive modeling development and evaluation.
- System deployment architecture with integrated suggestion capability.

3.2 Research Type and Design

3.2.1 Research Type

This research contains a quantitative research approach design with a focus on several types of numerical variables and quantifiable attributes. Quantitative method is appropriate as the study involves analysis of numerical IMDb ratings, quantifiable independent variables (budget, runtime, historical figures, embeddings), and requires statistical analysis of classification performance on objective metrics (accuracy, precision, recall, F1-score).

The research is exclusively founded on secondary data from open sources (Kaggle, IMDb, TMDb, Wikipedia) rather than the collection of primary data. This is reasoned in that pre-existing datasets provide validated data spanning decades, enable examination of big-picture trends across thousands of films, provide longitudinal data required for time-sensitive historical calculations, and enable research replicability through independent verification.

3.2.2 Research Design and Requirement Analysis

The research adopts a correlational-predictive design with binary classification as the main analysis type. The correlational component analyses relationships between pre-release attributes (independent variables) and performance categories (dependent variable), identifying attributes influencing success. The predictive component which primarily focus on developing, optimizing, and evaluating classification models to predict unseen movies into success categories.

3.2.2.a Target Variable Definition:

In this study, movie performance is operationalized as binary classification based on IMDb ratings:

- Successful: IMDb rating ≥ 6.5
- Failed: IMDb rating < 6.5

The 6.5 threshold was chosen after reviewing IMDb's own rating guidelines, which consider ratings above 7 as "good" and ratings between 6 and 7 as "not bad." Since the median of our dataset is 6.4, setting the cut-off at 6.5 balances the two classes reasonably while aligning with industry conventions. This binary classification is preferred over regression because it provides more actionable information for investment decisions, prioritizes failure avoidance for stakeholders, and reduces sensitivity to fluctuations in mid-range ratings[19].

3.2.2.b Stakeholder Identification

The system is conceptualized as a decision-support tool for multiple stakeholder groups within the movie-making ecosystem:

- Film producers and studio executives: Direct users who take go/no-go decisions on project greenlighting and film development.

- Marketing and distribution teams: Tactical planners who require lead indicators of probable performance for campaign planning and resource allocation.

- Financial stakeholders: Investors and finance analysts evaluating risk profiles and commercial potential of film portfolios.

3.2.2.c Functional Requirements Specification

To satisfy stakeholder and research needs, the following core functional requirements were established:

- Multi-Attribute Data Input: The system must accept full pre-release movie attributes like budget, running time duration, genre classification, MPAA rating, past performance data of top talent (leading actors and director), and detailed plot synopsis.

- Binary Classification Output: Upon processing input data, the system must generate a clear binary prediction ("Success" or "Unsuccess") and a calibrated probability score reflecting prediction confidence.

- Contextual Recommendation Generation: For each analyzed movie, particularly those predicted as unsuccess, the system must provide intelligent recommendations through an AI-based assistant capable of:

  1. Retrieving analogous successful movies from historical data.
  2. Generating specific, actionable improvement suggestions through comparative analysis.

- Interactive User Interface: The system must possess a simple web-based interface so that non-technical users can input film properties and interpret results without machine learning expertise.

3.2.2.d Non-Functional Requirements

In keeping with real-world usability and user acceptance, the following quality attributes were established:

- Predictive Reliability: The classification model must demonstrate well-balanced performance in terms of precision and recall with an F1-score of at least 0.75 to qualify for production.

- System Responsiveness: The deployed application should respond quickly to make predictions and recommendation responses efficiently to ensure decent user experience.

- Interpretability: System outputs should be clear and actionable to the non-technical stakeholders, probability scores and comparative recommendations enhancing decision confidence.

- Scalability: The system must cope with concurrent user requests and accommodate future growth in the dataset without loss of performance.

3.2.2.f Technical and Tools Requirements

The entire machine learning pipeline was created in Python 3.9, leveraging a suitably selected technology stack:

- Core ML Framework: scikit-learn 1.0 (preprocessing, baseline models, ensemble utilities)
- Gradient Boosting: XGBoost 1.5, LightGBM 3.3
- Deep Learning: PyTorch 1.10, Hugging Face Transformers 4.18
- Numerical Computing: NumPy 1.21, Pandas 1.3
- Similarity Search: FAISS 1.7 (Facebook AI Similarity Search)
- API Framework: FastAPI 0.75
- Deployment: Uvicorn ASGI server

Development was conducted on a workstation with NVIDIA GPU support for accelerated transformer inference, although final deployment utilized CPU-only inference for accessibility.

3.2.2.e Research Scope

The primary scope involves the construction, optimization, and testing of machine learning classification models for accurate movie performance prediction using only pre-release data. The compound deployment objective incorporates a content-based recommendation engine that provides similar successful movies, with comparative benchmarks as a supplementary deployment function.

3.3 Data Collection and Integration

3.3.1 Data Sources and Tools

The analytical dataset was constructed via a multi-stage process of collecting and merging data from multiple individual, high-quality sources:

- Structured Metadata: The foundation dataset was created from a set of year-wise metadata files from Kaggle, ranging from 2000 to 2023. The files provided an exhaustive set of initial features for every movie, including Title, Year, Duration, MPA, Rating, budget, directors, stars, genres, countries_origin, and Languages as shown in Table 3.1.

Table 3. 1: Meta data collection from Kaggle

Variable	**Description**
Title	The official name of the movie.
Movie Link	A direct URL to the movie's IMDb page.
Year	The release year of the movie.
Duration	The total runtime of the movie, usually in hours and minutes.
MPA	The Motion Picture Association rating indicating age suitability (e.g., PG, R).
Ratings	The average IMDb user rating for the movie.
Votes	The total number of user ratings submitted on IMDb.
budget	The estimated production budget of the movie.
grossWorldWide	The movie's total box office revenue worldwide.
gross_US_Canada	The box office revenue from the U.S. and Canada.
opening_weekend_Gross	Revenue earned during the opening weekend in the domestic market.
directors	Names of the directors responsible for the movie.
writers	Names of the writers or screenwriters involved.
genres	Lead actors or main cast featured in the movie.
countries_origin	The countries where the movie was produced.
filming_location	The locations where the movie was filmed.
production_companies	The companies involved in producing the film.
Languages	The spoken languages in the movie.
wins	The number of awards the movie has won.

nominations	The total number of award nominations received.
oscars	The number of Academy Awards (Oscars) the movie has won.

- Historical Performance Data: To engineer dynamic actors and director average IMDB rating, more data was programmatically collected by using two key APIs. The Movie Database (TMDb) API was used to retrieve the complete filmographies of directors and actors. The Open Movie Database (OMDb) API and the IMDbPY library were then used to retrieve the precise IMDb ratings for all the relevant historical films contained within the filmographies.

- Plot Narrative Data (Unstructured): The unstructured text data in the form of plot narratives were systematically crawled from Wikipedia. This was accomplished using a custom Python script, which employed the Wikipedia API for efficient searching and section extraction, and the BeautifulSoup library for parsing HTML pages. Wikipedia was used due to its generally detailed and elaborate narrative descriptions, most appropriate for semantic analysis.

3.3.2 Data Selection and Initial Filtering

Prior to integration, the raw data were subjected to a strict selection and filtering procedure in order to produce a concentrated and consistent dataset for modeling;

- Initial Feature Selection: A subset of relevant pre-release features was selected from the comprehensive set of columns available in the Kaggle datasets. Columns corresponding to post-release outcomes (grossWorldWide, wins, nominations, etc.) were removed to maintain temporal validity.

- Temporal Scope: The records were restricted to films released between the year 2000 and 2023. Films released from the year 2024 and beyond were excluded because IMDb ratings take from 12-18 months to settle, ensuring that the target variable (Rating) for every film in the dataset is a stable, long-term measure of audience acceptance [20].

- Geographic and Language Focus: In order to control for market and culture-specific factors, the data was limited to English-language films produced in America. This was accomplished by filtering records where countries_origin included the United States and the Languages column indicated English.

- Genre and Format Exclusions: In order to focus the model's training on theatrical feature films, records that portrayed documentaries, TV series, mini-films, short videos, and animations were systematically excluded using information in the genre’s column.

- Talent Extraction: The key creative talent was extracted from list-containing fields in each movie. The primary director was set as the first director’s name listed in the director’s column. Similarly, the first two names listed in the star’s column were selected as the lead and second lead actors. The first genre listed in the genre’s column was retained as the primary genre of the movie.

3.3.3 Data Integration Workflow

The filtered and chosen data from the heterogeneous sources were integrated systematically into a single unified analytical dataset:

- Base Dataset Creation: The cleaned and filtered metadata from the Kaggle datasets formed the base, where every row represented a single movie.

- Integration of Historical Metrics: The Python scripts were executed to calculate the time-sensitive historical average ratings for the director and two leads of each movie that were scraped. These additional numerical features were joined back into the base data.

- Matching Plot Synopsis: The Wikipedia scraping script was run to gather plot synopses for each film, with Title and Year serving as primary identifiers for search. The script employed a sophisticated search query technique, prioritizing queries like "Title (Year film)" to get the most likely Wikipedia page and falling back on methods to extract raw HTML if section extraction via API failed. The plot text so extracted was matched and consolidated with the corresponding movie in the dataset.

3.4 Data Preprocessing and Feature Engineering

3.4.1 Data Cleaning and Standardization

Following integration, the data was cleaned and normalized lastly to prepare it for modeling:

- Budget: All the budget values were normalized by removing the dollar sign and representing them as plain numeric values for consistency and simplicity in comparison across the 24-year period.

- Runtime: All the values for duration were changed to the same integer format for total minutes.

- Missing Values: In the budget numeric column, missing values were imputed through the median budget of films within the same top genre category (genre-stratified median imputation). In the MPA rating categorical column, missing values were explicitly assigned an "Other" category. Records remaining missing critical identifying information after initial filtering (i.e., director, leading cast, or plot) were excluded from the final dataset to prevent the introduction of systematic bias.

3.4.2 Feature Engineering: Time-Sensitive Historical Performance Metrics

One of the methodological innovations of this research is the feature engineering of time-sensitive historical performance metrics for actors and directors, avoiding anachronistic data leakage:

- Methodology: For every film released in year Y, the previous-year average IMDb rating of its director and principal actors was computed using only films they had released in years prior to Y. This ensures the metric actually reflects the talent's pre-existing reputation at the time pre-production decisions were being made.

- Implementation Process: This was deployed via a multi-step Python script that automated a complex data retrieval workflow:

    1. For each director and actor, the TMDb API was queried to obtain their person ID.

    2. This ID was used to retrieve their complete filmography.

    3. The filmography was then filtered based on strict conditions: the release year of the movie was required to be less than year Y, it had to be a U.S. production, its running time was required to be at least 40 minutes, and it could not be of an excluded genre like "Documentary."

    4. For all films that passed these filters, their IMDb ratings were retrieved using the OMDb API or IMDbPY library.

    5. An arithmetic mean of these ratings was calculated to produce the final dynamic historical average.

- Technical Implementation: The procedure was rendered as efficient as it could be. Caching was implemented to store already fetched person IDs and ratings so that unnecessary API calls were avoided. For actor ratings, Python's concurrent.futures.ThreadPoolExecutor was employed to process multiple actors concurrently, which significantly accelerated the data gathering procedure.

3.4.3 Categorical Variable Encoding

Nominal categorical attributes were converted into a numerical representation suitable for machine learning algorithms through one-hot encoding. One-hot encoding transforms each category into a separate binary feature, where a value of 1 indicates the presence of that category and 0 indicates its absence. This allows algorithms that require numerical input to process categorical data without implying any ordinal relationship between categories. To prevent creating an excessively high-dimensional and sparse feature space, a consolidation strategy was implemented before encoding:

- MPAA Ratings: The attribute was reduced to the most frequent classes, "PG-13" and "R," and all other ratings (i.e., G, PG, NC-17) were grouped together into one "Other" category.

- Genres: The primary genre was reduced to the four most common forms "Action," "Drama," "Comedy," andv "Biography" and all the remaining genres were included in an "Other" category.

OneHotEncoder from the scikit-learn library was then used to transform these reduced features into binary columns, with the drop='first' parameter being used to prevent multicollinearity.

3.4.4 Plot Synopsis Preprocessing and Embedding

To be able to include the narrative content of both movies in the model, the unstructured plot summaries were transformed into dense numeric vectors (embeddings):

- Text Cleaning: To clean the plot summary specialized steps were runover each and every plot summary. And this process contains with multiple steps of processes: decoding HTML entities, stripping Wikipedia-specific artifacts such as [edit] tags and citation numbers, removing all URLs and HTML tags, and normalizing whitespace and punctuation to leave a clean, pure narrative text.

- To get the embeddings for all plot summaries, the model that were used is Longformer model that names as allenai/longformer-base-4096. It was the final model that selected particularly for this task. This model creates longer context window with 4096 tokens compares to BERT models that have 512 tokens, this Longformer model has the capability to accommodate the full synopsis which frequently surpasses 512 tokens and to be processes without any truncation, thus maintaining the full narrative structure.

- Generation Process: After get the cleaned plot text, it fed into the Longformer model (allenai/longformer-base-4096). As a first step the text was tokenized by the tokenizer of the model and padded up to the maximum length of the model that is

4096. This was preceded by a forward pass through the model, followed by mean pooling over the final hidden layer of token embeddings. This process, which properly considers padding using the attention mask, pools the context-sensitive token information into a fixed-size 768-dimensional representation of the semantic gist of the entire plot.

3.4.5 Dimensionality Reduction

The resulting 768-dimensional embedding space was dense and high-dimensional. With the aim of achieving a stronger and computationally lighter feature set, dimensionality reduction was carried out:

- Standardization: The embeddings were standardized to have zero mean and unit variance with scikit-learn's StandardScaler.

- Variance Analysis: For the selection of components, Truncated Singular Value Decomposition (SVD) was trained on the range of component values (20 to 300). Cumulative explained variance was graphed against the number of components to determine the inflection point.

- Component Selection: Analysis proved that 82 significant factors accounted exactly for 80% of the total variance in the embedding space. This threshold is an ideal amount of information retention and overfitting prevention, as established in cross-validation tests.

- Final Transformation: Truncated SVD with n_components=82 reduced the 768 dimensions to 82 principal components. The transformation retained about 80% of explained variance, preserving much of the valuable semantic information without the danger of overfitting and removing potential noise in the lower-variance dimensions.

3.5 Exploratory Data Analysis Approach

A comprehensive exploratory data analysis (EDA) was conducted to ascertain the underlying structure of the data set, identify patterns, and guide subsequent methodological choices about preprocessing and modelling. This was an exercise in reconciling statistical summaries with data visualization for research design.

- Descriptive Statistics and Distribution Analysis: The initial process was the computation of descriptive statistics (mean, median, standard deviation, skewness, kurtosis) for the quantitative variables. The histogram was then employed in graphically representing each of the distributions. This was carried out so as to search for salient features such as central tendency, dispersion, and skewness. This was an important step in identifying variables to be transformed, e.g., using a logarithm transformation for very skewed data, in order to further meet the assumptions of certain modelling methods.

- Target Variable and Class Balance Analysis: After operationalizing the continuous Rating variable into a binary target class ("Successful" vs. "Unsuccessful"), frequency distribution analysis was conducted. This was with the view to determine the proportion between the two classes. This was significant in informing the choice of a stratified sampling method when dividing data in a way that proportional representation of all classes would be maintained for training and testing sets to prevent biased model estimation.

- Correlation and Bivariate Analysis: A Pearson correlation matrix was computed and represented as a heatmap to test correlations between numerical features and the target feature. This was used to assess direction and strength of linear associations and to test prediction capability of features obtained through derivation. Scatterplots were similarly produced to look at visually trends in-between Rating variable and most vital predictors, a method employed to detect any non-linear trends which would not be detected by correlation coefficients.

- Categorical Feature Analysis: The impact of categorical variables, such as genre and MPAA rating, was examined through the use of boxplots and count plots. Boxplots were utilized to compare the distribution of the Rating variable within each category to be able to visually assess median value differences and variance. Stacked bar plots were also created to illustrate proportion of successful and failed films for each category. A feature analysis of production_companies was also conducted to determine its cardinality and correlation with the target variable. This was needed to enable us to make logical choices in feature removal to prevent model overfitting and potential noise.

- Narrative Content Analysis: The plot synopsis data was analysed to learn about its text characteristics and its behaviour.

- Synopsis Length Analysis: The distribution of plot length was quantified in terms of word length and token length per entry. Token length was quantified through a BERT tokenizer to obtain an exact estimation of the input size that transformer-based models would require. This was to justify the selection of an optimal embedding model in terms of context window size.

- Correlation Analysis Embedding: After plot embedding generation and dimensionality reduction, correlation analysis between the derived SVD components and Rating target variable was conducted. The purpose of this step was to ensure that the abstracted narrative features contained predictive signal relative to the research question and therefore should be included within the final feature set.

These exploratory processes were instrumental in shaping the final approach. They directly impacted significant preprocessing decisions, drove feature selection, and affected the choice of modeling strategies so that they were best aligned with the specific characteristics of the data.

3.6 Predictive Modeling and Optimization

3.6.1 Model Selection Rationale

In the initial stage of model selection, numerous classification algorithms were trained and tested in order to identify the best predictors of the success of a movie. This exploratory phase involved the use of traditional machine learning models such as Random Forest, Support Vector Machines (SVM), and K-Nearest Neighbors (KNN). While these models served to give baselines for performance that measured by f1-score, the study determined that gradient boosting techniques were superior for predictive reliability and accuracy on this specific data set. Consequently, the final production model was developed using a more sophisticated ensemble method in an effort to maximize performance.

The final production model that developed in this research employs a Regularized Stacking Ensemble framework incorporating two state-of-the-art gradient boosting techniques:

- XGBoost (Extreme Gradient Boosting): Selected as a base learner due to its state-of-the-art performance on structured tabular data. XGBoost employs sequential ensemble learning with each decision tree that corrects errors of predecessors via gradient descent optimization. Its most important advantages are native support for complex non-linear interactions of features, built-in L1/L2 regularization against overfitting, efficient parallel processing capabilities, and the capacity to support class imbalance via the scale_pos_weight parameter.

- LightGBM (Light Gradient Boosting Machine): Selected as a complementary base learner providing alternative gradient boosting implementation with faster training by utilizing histogram-based algorithms and leaf-wise tree growth. LightGBM's alternative tree-building strategy (leaf-wise rather than XGBoost's level-wise) produces distinct error patterns that enhance ensemble diversity.

- Logistic Regression with Cross-Validation (LogisticRegressionCV): Selected as the meta-learner (ultimate estimator) for the stacking ensemble. The meta-model learns optimal weighted averages of predictions from base learners through logistic regression, with the regularization strength automatically selected based on 5-fold cross-validation. The use of Logistic Regression provides interpretable probability calibration and prevents overfitting in the meta-learning process.

- Architecture Rationale: The ensemble stacking architecture leverages the complementary strengths of XGBoost and LightGBM—stable base predictions from XGBoost's shallow, regularized trees, and intricate interaction capture from LightGBM's aggressive leaf-wise growth. The LogisticRegressionCV meta-learner weights these diverse predictions optimally while providing probability calibration through internal cross-validation.

3.6.2 Data Partitioning and Preprocessing

Dataset Split: The whole dataset (1,400 films) was split into training and test sets on an 80/20 basis with stratified sampling to preserve class ratios:

- Training Set: 1,120 films (80%)
- Test Set: 280 films (20%)
- Stratification: Maintained 51-49 Success-Unsuccess ratio in both sets
- Random Seed: Fixed to 42 for reproducibility

Feature Scaling: StandardScaler was applied only to the BERT embeddings before SVD transformation to attain zero mean and unit variance. Tree models (XGBoost, LightGBM) trained on the final feature set do not require feature scaling because decision trees split by threshold comparisons that are scale-invariant by nature.

Final Feature Configuration:

- Numerical Features: budget, Duration_Minutes, First Actor Avg, Second Actor Avg, Average IMDb Rating (5 features)
- Categorical Features (One-Hot Encoded):
- MPA: "PG-13", "R", "Other" → 2 binary features
- Genre: "Action", "Drama", "Comedy", "Biography", "Other" → 4 binary features
- Narrative Features: 82 SVD-reduced plot embedding dimensions (bert_svd_0 to bert_svd_81)
- Total Feature Dimensionality: 93 features (5 numeric + 6 categorical + 82 embedding)

3.6.3 Hyperparameter Tuning using RandomizedSearchCV

To identify the optimal model configuration, RandomizedSearchCV was employed with extensive hyperparameter search over both base estimators. This approach balances exploration breadth and computational efficiency by sampling parameter configurations from specified distributions rather than exhaustive grid search.

Search Configuration:

- Search Strategy: RandomizedSearchCV with 50 iterations

- Cross-Validation: 3-fold stratified CV
- Scoring Metric: f1_weighted (harmonic mean of precision and recall, weighted by class support)
- Parallel Processing: n_jobs=-1 (utilizing all available CPU cores)
- Random Seed: 42 for reproducibility

XGBoost Parameter Space:

- n_estimators: [50, 100, 200, 400] (number of boosting rounds)
- learning_rate: log-uniform(0.01, 0.3) (continuous distribution)
- max_depth: [3, 4, 6] (maximum tree depth)
- gamma: log-uniform(0.01, 1.0) (minimum loss reduction for split)
- reg_alpha: log-uniform(0.01, 10.0) (L1 regularization)
- reg_lambda: log-uniform(0.01, 10.0) (L2 regularization)
- subsample: uniform(0.6, 1.0) (sample fraction per tree)
- colsample_bytree: uniform(0.6, 1.0) (feature fraction per tree)

LightGBM Parameter Space:

- n_estimators: [50, 100, 200, 400]
- learning_rate: log-uniform(0.01, 0.3)
- num_leaves: [31, 50, 100] (maximum leaves per tree)
- min_child_samples: [10, 20, 30] (minimum samples per leaf)
- reg_alpha: log-uniform(0.01, 10.0) (L1 regularization)
- reg_lambda: log-uniform(0.01, 10.0) (L2 regularization)
- feature_fraction: uniform(0.6, 1.0) (feature sampling ratio)
- bagging_fraction: uniform(0.6, 1.0) (data sampling ratio)
- bagging_freq: [1, 5, 10] (bagging frequency)

3.6.4 Final Ensemble Architecture

The production model employs a two-layer stacking ensemble architecture code is shown in Appendix A:

Layer 1 - Base Estimators:

- Optimized XGBoost Classifier: Shallow trees (depth=3) and moderate regularization, precision-oriented through conservative predictions
- Regularized LightGBM Classifier: Larger ensemble (400 trees) and strong L1 regularization, recall-optimized through aggressive feature selection

Layer 2 - Meta-Learner:

Logistic Regression with Cross-Validation (LogisticRegressionCV):

- Cross-validation: 5-fold
- Regularization parameter search: 10 values (Cs=10)
- Max iterations: 1000
- Trained on base estimators' probability predictions made through 5-fold cross-validation

Architecture Rationale:

- Diversity: XGBoost and LightGBM have different tree construction algorithms (depth-wise vs. leaf-wise), leading to diverse error patterns that can be exploited by the meta-learner.
- Complementarity: Shallow trees in XGBoost provide stable base predictions, while aggressive feature selection in LightGBM captures complex interactions.
- Meta-Learning: LogisticRegressionCV automatically learns optimal regularization strength, with interpretable probability calibration and optimal base prediction weighting.
- Regularization Pyramid: Aggressive regularization at both base level (L1/L2 in XGBoost/LightGBM) and meta level (LogisticRegressionCV) yields an overfitting-resistant ensemble.

3.6.5 Training Protocol

Cross-Validation Strategy:

- 5-fold stratified cross-validation in the StackingClassifier ensures base learner predictions for training the meta-learner are made via cross-validation, to prevent overfitting
- 3-fold stratified cross-validation during RandomizedSearchCV for hyperparameter tuning

- 5-fold cross-validation in LogisticRegressionCV for meta-learner regularization strength selection

Model Training Process:

- RandomizedSearchCV searches 50 parameter combinations over 3-fold CV (150 total fits)
- Best parameter configuration chosen based on f1_weighted score
- Final model refitted on whole training set with best parameters
- Base estimators make out-of-fold predictions through 5-fold CV for meta-learner training
- LogisticRegressionCV fits on base predictions with automatic regularization selection
- Probability Threshold: The default classification threshold (0.5) was retained for simplicity and balance in the balanced performance of the two classes. Post-training threshold tuning was not performed since the model exhibited balanced precision-recall at the default threshold.

3.6.6 Model Evaluation Metrics

Comparative evaluation measured the stacking ensemble on the following complementary metrics:

- Accuracy: Overall percentage correctness
- Precision: True positive rate within positive predictions (investment quality)
- Recall: True positive rate within actual positives (opportunity capture)
- F1-Score: Harmonic mean of precision and recall (primary selection criterion)
- Classification Report: Per-class precision, recall, F1-score, support
- Learning Curves: Training and validation F1-scores for different training set sizes to assess convergence and overfitting

The XGBoost-LightGBM stacking ensemble with probability calibration using LogisticRegressionCV and tuned hyperparameters was selected as the final deployment model based on superior F1-score, balanced precision-recall, and excellent generalization demonstrated via stratified cross-validation.

3.7 System Deployment with Integrated Recommendation

3.7.1 Deployment Architecture

The prediction system is realized as a high-performance web service using the FastAPI framework, providing a robust and scalable API for real-time prediction and analysis. The infrastructure is designed for efficiency and reproducibility, and all components of the modeling pipeline are encapsulated for production deployment.

Artifact Serialization: Deployment revolves around serialized artifacts. The final, best-performing model, a stacking classifier and all preprocessing components necessary are saved using joblib:

- STACKED_MODEL.joblib: Complete trained StackingClassifier with optimized XGBoost and LightGBM base estimators
- ONE_HOT_ENCODER.joblib: Fitted OneHotEncoder for categorical features (MPA, Genre)
- BERT_SCALER.joblib: Fitted StandardScaler for plot embedding normalization before SVD
- SVD_TRANSFORMER.joblib: Fitted TruncatedSVD with n_components=82 for dimensionality reduction
- TARGET_ENCODER.joblib: Fitted LabelEncoder for target variable (Success/Unsuccess)
- feature_names.joblib: Ordered list of 93 feature names for alignment at prediction time

Application Lifecycle Management: On application startup, a separate process pre-loads all the artifacts required into memory:

- The serialized machine learning model and preprocessing pipelines
- The pre-trained allenai/longformer-base-4096 model and tokenizer from the Hugging Face library, which is required for generating plot embeddings on the fly
- Big KnowledgeBase of 571 previous successful films (IMDb ≥ 6.5) that preloads a FAISS index for quick similarity queries

This preloading strategy saves latency on prediction and chat requests because the data structures and models are already in memory.

API Endpoints:

- Prediction Endpoint (/predict): Accepts a JSON payload of a movie's pre-release features (budget, running time, talent scores, plot summary, etc.). Executes the complete end-to-end prediction pipeline:
  1. Raw plot text is preprocessed with custom text preprocessing functions
  2. Preprocessed text is encoded into 768-dimensional embedding using the Longformer model
  3. Embedding is scaled with bert_scaler and reduced to 82 dimensions using SVD
  4. Categorical features (MPA, Genre) are normalized and one-hot encoded
  5. Numerical features are concatenated with encoded categoricals and SVD embeddings
  6. Complete 93-dimensional feature vector is passed to the trained stacking model
  7. Returns JSON response with binary classification ("Success" or "Unsuccess") and probability score

- Chat Endpoint (/chat): Powers the interactive AI assistant, accepting a user's message and the context of the movie being analyzed. Runs the recommendation engine workflow:
  1. Generates composite embedding for input movie (metadata + plot)
  2. Performs FAISS similarity search to obtain top-5 similar successful movies
  3. Constructs context-aware prompt with similar movies' metadata and plots
  4. Invokes Groq API (GPT-OSS-20B model) to generate natural language recommendations
  5. Returns AI-generated response along with similar movies list and actionable insights

- Health Endpoint (/health): Standard health check endpoint for monitoring API operational status and verifying that all the components, including the knowledge base, are loaded correctly.

3.7.2 Suggestion Integration

To supplement the primary classification, the system includes a sophisticated Suggestion and analysis engine, delivered through an interactive AI chatbot. And the objective of this chatbot is to provide real world applicable and actionable suggestion system to this overall system deployment. By comparing an input movie to a hand-curated knowledge base of previous successes, the system allows for the identification of key differentiators and potential areas for improvement.

.

Knowledge Base and Similarity Search:

- Knowledge Base: The knowledge base contains with the 678 successful movies that have IMDB rating equal to or above 6.5 from whole dataset that contains 1400 movies and in the knowledge base there are meta data and 768 dimensional Longformer embeddings.

- FASS Index: Constructed using IndexFlatL2 (exact L2 distance computation) from the 768-dimensional embeddings, enabling sub-millisecond similarity searches.

- Similarity Metric: L2 (Euclidean) distance in the 768-dimensional embedding space.

- Implementation using AI Chatbot: When a user asks for suggestions, the system performs the following:
  1. Creates a single embedding for the user's input movie by combining its numerical, categorical, and narrative features into a single 768-dimensional vector representation
  2. This query vector is used to search the FAISS index, returning the top-5 most similar successful movies from the knowledge base based on L2 distance
  3. The plot summaries and metadata of these similar movies are gathered as rich context
  4. This context, along with the user's exact question, is input to a large language model (Groq API with GPT-OSS-20B) which combines this information to generate coherent, natural language responses

Rationale: This integration through chatbot is incredibly valuable relative to a simple list of suggestions. It allows stakeholders to ask targeted questions and receive data-driven answers around:

- Comparing their project's budget, runtime, or talent ratings to successful benchmarks

- Uncovering common narrative themes or plot structures of similar successful films

- Receiving synthesized, data-driven suggestions for potential improvements to their project

- This interactive presentation makes the system an effective decision-support tool, enabling dynamic and exploratory analysis of a film's commercial potential.

3.8 Ethical Considerations

- All data used in this study were publicly accessible from Kaggle, IMDb, Wikipedia, and TMDb.
- No sensitive or personal data were collected or processed at any stage of the research.
- Full compliance with platform-specific non-commercial research terms (e.g., IMDb) and Creative Commons licensing (Wikipedia).
- Rate limits and usage policies of the platforms were adhered to in all API use.
- Transparency was maintained across the data sources, preprocessing methods, and
- methodological limitations.
- Potential biases were acknowledged, including:
  1. Genre representation bias
  2. Temporal trends in movie performance
  3. User demographic impacts on ratings
- The predictive system is a decision-support tool and not a deterministic authority.

Final decisions are expected to be made with human judgment and domain expertise to ensure responsible use of the system.

3.9 Summary

This framework offers a rigorous approach to binary film performance categorization from pre-release traits and narrative content. The primary objective is to precise categorization into success types and it is achieved by strict data fusion, novel time-aware historical features, transformer-based narrative embeddings, widespread model hyperparameter fine-tuning with RandomizedSearchCV, and stringent evaluation.

The deployment design incorporates a content-based recommendation engine powered by FAISS similarity search and LLM-generated contextual advice as an integrated complementary function enhancing practical utility through similarity-based benchmarking.

CHAPTER 4
RESULTS AND EVALUATION

4.1 Overview

Chapter 4 describes the overall findings from the development and testing of the suggested movie performance prediction and recommendation system. Outcomes of the research span various phases from exploratory data analysis to model deployment, system architecture, and empirical testing. The chapter follows in a coherent path the analytics journey from requirements elicitation at the start to deployment and testing of a functional decision-support system.

The chapter is organized in four interdependent sections which collectively describe the project's whole life cycle:

- Section 4.2 - Data Exploration and Domain Knowledge: This section provides crucial findings that identify through the data exploration and domain knowledge phase that impacted feature engineering and modeling techniques.

- Section 4.3 - Model Building and Implementation: This section outlines the process of building the model iteratively from starting algorithms right through to the end ensemble design.

- Section 4.4 - Testing and Validation: Offers quantitative measures of performance and functional testing results that validate the system's performance.

Each of the following sections encompasses the preceding one, with a unifying narrative that binds analysis conclusions, design decision, implementation options, and ultimately, system performance results.

4.2 Exploratory Data Analysis

4.2.1 Dataset Composition and Characteristics

Analytical dataset is a carefully curated collection of English language, American-made films from 2000 to 2023. By merging data from several sources like IMDB, TMDB, and Wikipedia gives comprehensive dataset with structured and unstructured data.

- Initial Dataset: Over 6,000 films

- Final Analytical Dataset: 1,400 films (under strict quality filtering and completeness verification)

The dataset structure combines four categories of data:

- Quantitative Attributes: Continuous numeric attributes like production budget and runtime length

- Categorical Attributes: Nominal attributes such as genre type, MPAA rating, and production studio

- Temporal Performance Metrics: Time-sensitive historical averages for lead performers and director, calculated to prevent data leakage

- Unstructured Text: Generous plot synopses from brief summary and description about the summary of narrative context.

4.2.2 Target Variable Analysis

Target variable of IMDb Rating is taken as the reference for binary classification. Statistical analysis yields important features regarding reception behaviour among audiences:

Characteristics of Distribution:

- Mean Rating: 6.37
- Standard Deviation: 0.92

- Skewness: -0.59 (weak negative skew)
- Minimum: 1.9
- Maximum: 9.0

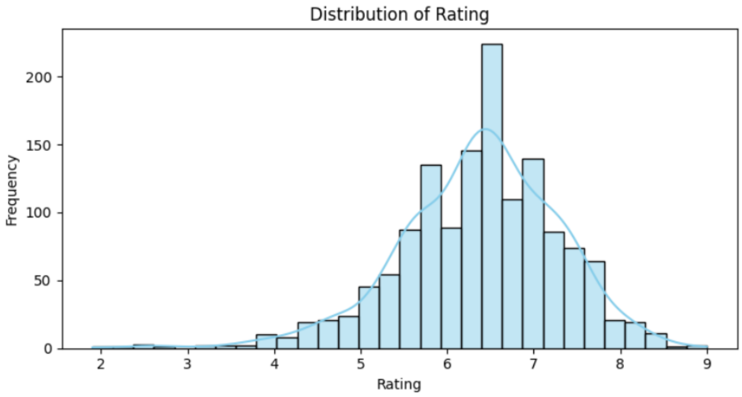


Figure 4. 1: *Distribution of rating*

The skewness indicates that the distribution is longer to the direction of lesser ratings, but the majority of the films cluster in the range 6.0–7.0 and can be identified by analysing the Figure 4.1. The trend indicates that most films are fairly successful, while bad films are comparatively less frequent in the filtered set, likely due to the pre-screening of extremely low-budget or minuscule-release films.

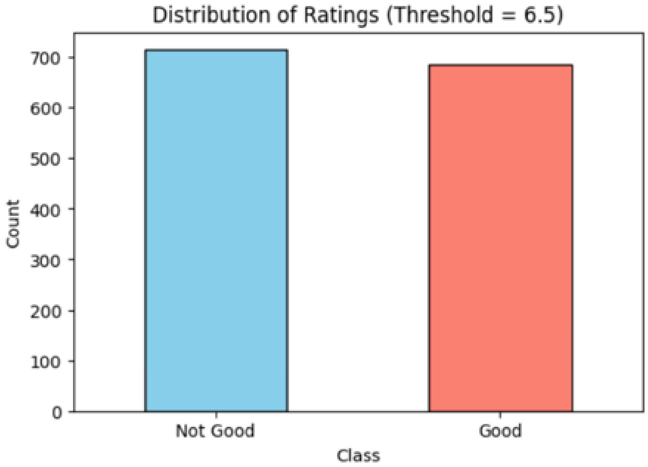


Figure 4. 2: Distribution of ratings

Binary Classification Threshold: On the basis of industry standards and distribution analysis, movies rated ≥ 6.5 on IMDb were labeled as "Successful" and the rest as "Unsuccessful," providing a relatively balanced dataset as depicted in Figure 1.2 with around 49% successful movies.

4.2.3 Numerical Feature Analysis

4.2.3.a Budget Distribution and Transformation

Production budget has features common in financial data in the entertainment industry:

- Mean: $56.7 million
- Median: $38.0 million
- Standard Deviation: $57.8 million
- Range: $0.1 million to $414.9 million
- Skewness: 2.05 (strong positive skew)
- Kurtosis: 5.37 (heavy-tailed distribution)

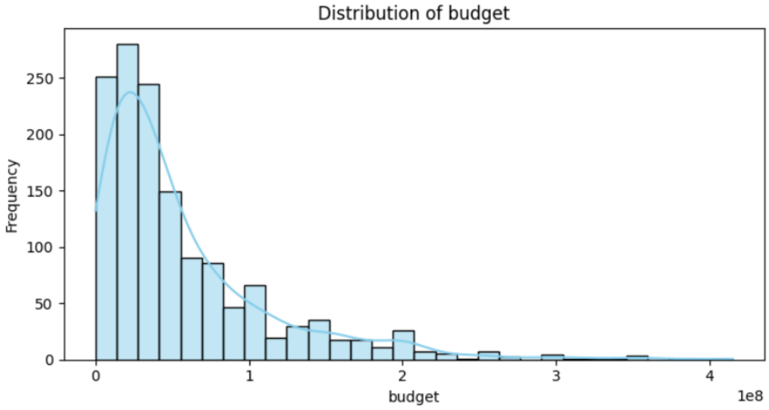


Figure 4. 3: Distribution of budget

The wide separation between mean and median, coupled with extreme positive skewness, confirm the presence of high-budget outliers while most of the films are concentrated in the low-budget ranges and also it can be observed in Figure 4.3. This type of skewness pattern needs logarithmic transformation during preprocessing to scale-normalize the feature for scale-sensitive modeling algorithms.

4.2.3.b Distribution of Duration

Movie durations in the data have the following characteristics:

- Mean: 113.1 minutes
- Median: 109.5 minutes
- Standard Deviation: 19.1 minutes
- Range: 73.0 minutes to 242.0 minutes
- Skewness: 1.17 (moderate positive skew)

- Kurtosis: 2.64 (slightly heavy-tailed distribution)

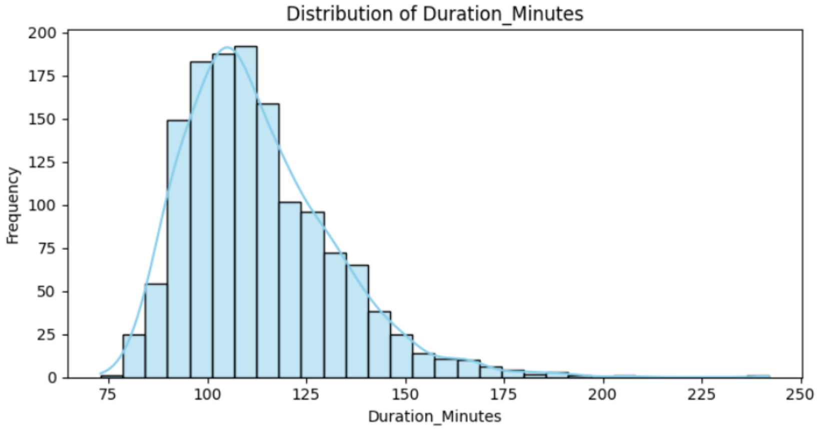


Figure 4. 4: Distribution of duration of a movie

By analysing the Figure 4.4 and the descriptive statistics, the positive skew indicates that even though most movies have lengths in the range of 100–120 minutes, there are some outliers that are far longer. This means that feature transformation or normalization can help to mitigate the effect of feature scale on performance-critical algorithms.

4.2.3.c First Actor Mean Rating

The average IMDb rating of the first actor indicates:

- Mean: 6.34
- Median: 6.37
- Standard Deviation: 0.48
- Range: 3.1 to 7.8
- Skewness: -1.09 (moderate negative skew)
- Kurtosis: 4.03 (heavy-tailed distribution)

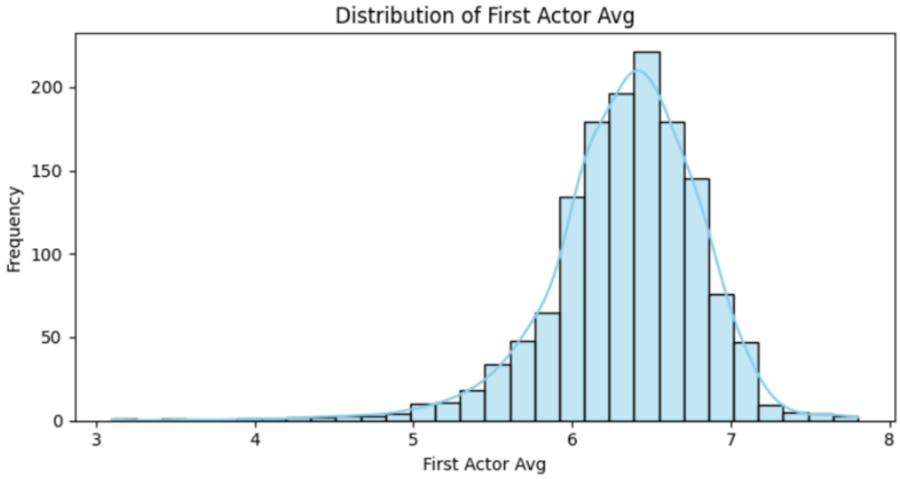


Figure 4. 5: Distribution of first actor average

As pointed out by the descriptive statistics and Figure 4.5, the negative skew indicates that the majority of first actors have ratings above the mean but some low-rated actors pull the distribution down. The trend may need standardization for modeling.

4.2.3.d Second Actor Average Rating

Second actor average IMDb rating shows:

- Mean: 6.24
- Median: 6.31
- Standard Deviation: 0.54
- Range: 2.7 to 8.0
- Skewness: -1.30 (moderate negative skew)
- Kurtosis: 4.74 (heavy-tailed distribution)

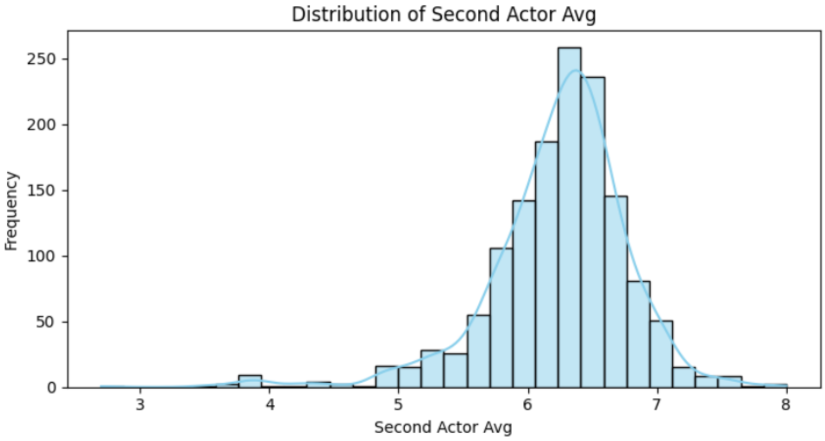


Figure 4. 6: Distribution of second actor average

According to the descriptive statistics and Figure 4.6, similar to the first actor, the second actor ratings are concentrated above the mean with several low-rated actors which create a leftward skew.

4.2.3.e Director Average Rating

Overall movie IMDb ratings are:

- Mean: 6.48
- Median: 6.52
- Standard Deviation: 0.77
- Range: 2.4 to 8.8
- Skewness: -0.71 (mild negative skew)
- Kurtosis: 2.05 (moderately heavy-tailed)

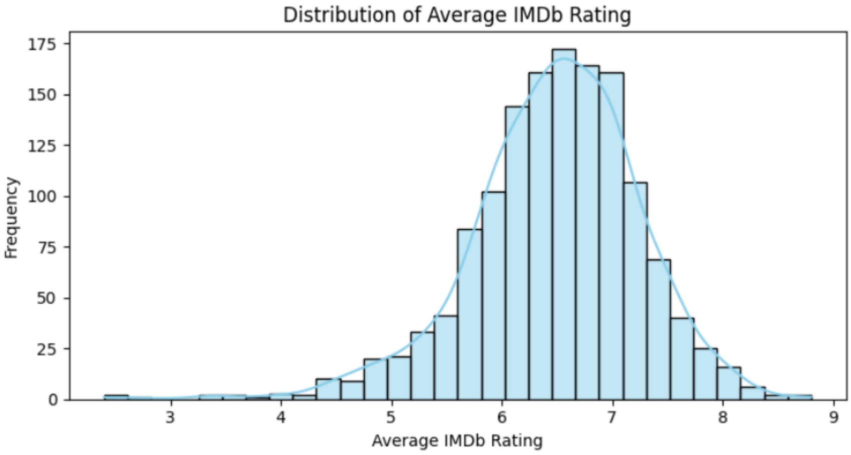
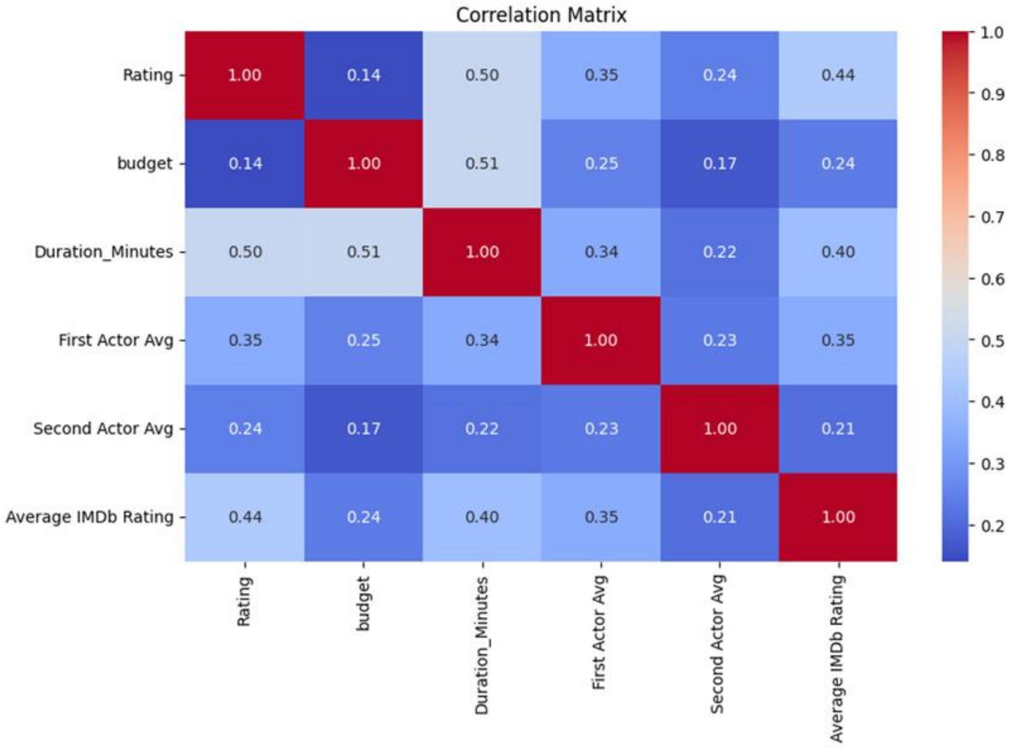


Figure 4. 7: Distribution of second actor average IMDB rating

From the descriptive analysis and Figure 4.7, the mild negative skew suggests that there are more movies with ratings higher than the mean, and a handful of lower-rated movies extend the lower tail. Minimal transformation may be required, but scaling will help with algorithms sensitive to scales on features.

4.2.4 Correlation Heatmap Analysis



Correlation Matrix

	Rating	budget	Duration_Minutes	First Actor Avg	Second Actor Avg	Average IMDb Rating
Rating	1.00	0.14	0.50	0.35	0.24	0.44
budget	0.14	1.00	0.51	0.25	0.17	0.24
Duration_Minutes	0.50	0.51	1.00	0.34	0.22	0.40
First Actor Avg	0.35	0.25	0.34	1.00	0.23	0.35
Second Actor Avg	0.24	0.17	0.22	0.23	1.00	0.21
Average IMDb Rating	0.44	0.24	0.40	0.35	0.21	1.00

Figure 4. 8: Correlation matrix

According to Figure 4.8, Moderate positive correlation exists between Rating and average actor scores (First Actor Avg, Second Actor Avg) and Average IMDb Rating, affirming their predictive power. Budget displays extremely weak correlation with Rating, demonstrating that greater expenditure does not automatically mean greater audience satisfaction. On the whole, the heatmap confirms the effectiveness of using actor performance and average IMDb ratings as major indicators of long-term box office success.

4.2.5 Categorical Feature Analysis

4.2.5.a MPAA Rating Distribution

MPAA rating distribution shows high intensity in mature-audience ratings:

- PG-13: 660
- R: 603
- PG: 130
- Other (G, NC-17, Not Rated): 7

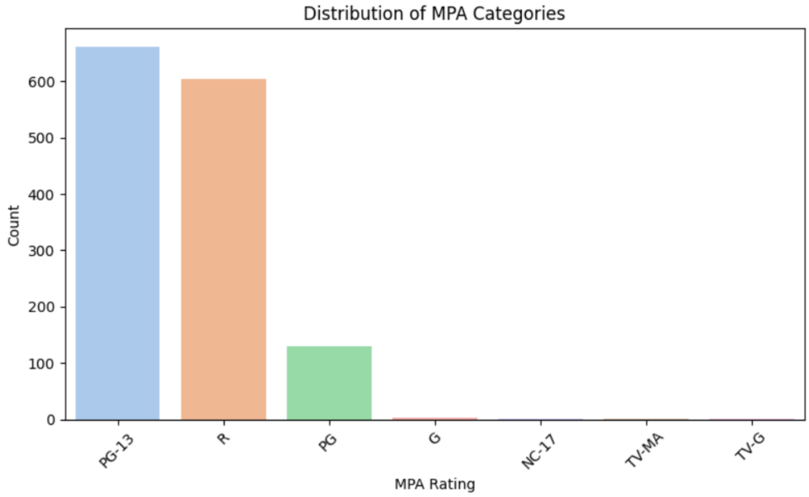


Figure 4. 9: Distribution of MPA categories

Both R and PG-13 are the most common ratings, which can be seen in the Figure 4.9, and denote a trend towards teen- and adult-oriented material. Child-friendly ratings like G, TV-G, and PG are proportionally few, which reflects lower demand for child cinema in current US film. Only fewer than ten or so have the ratings NC-17 or TV-MA, possibly because they are of little commercial value or have more stringent control over content.

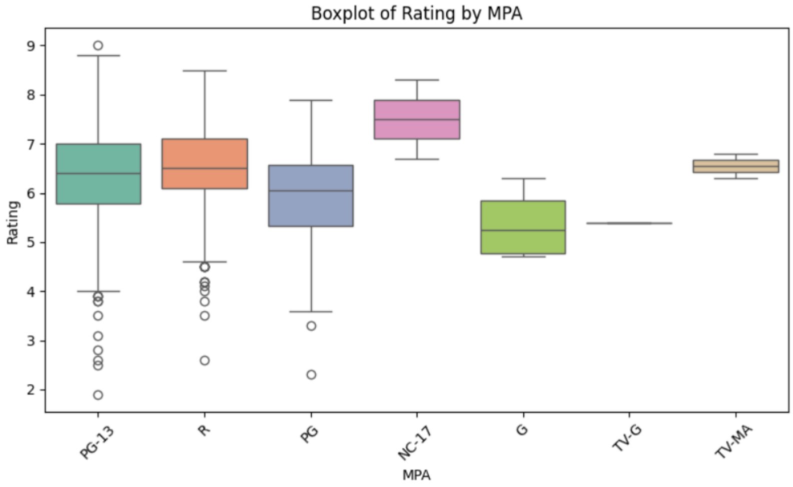


Figure 4. 10: Boxplot of rating by MPA

As illustrated in the Figure 4.10, mean ratings do vary slightly by MPAA category, PG-13 being 6.4 and R-rated films with a bit higher to the 6.4 doing equally well, the overlapping confidence intervals mean MPAA rating alone is not a good predictor of success. However, its inclusion as part of the feature set does provide valuable information when combined with other features like genre and budget.

4.2.5.b Genre Distribution and Performance Trends

Genre analysis shows concentration and unique performance patterns:

Frequency Distribution:

- 491: Action
- 362: Comedy
- 196: Drama
- 100: Biography
- 94: Crime
- 157: Other genres

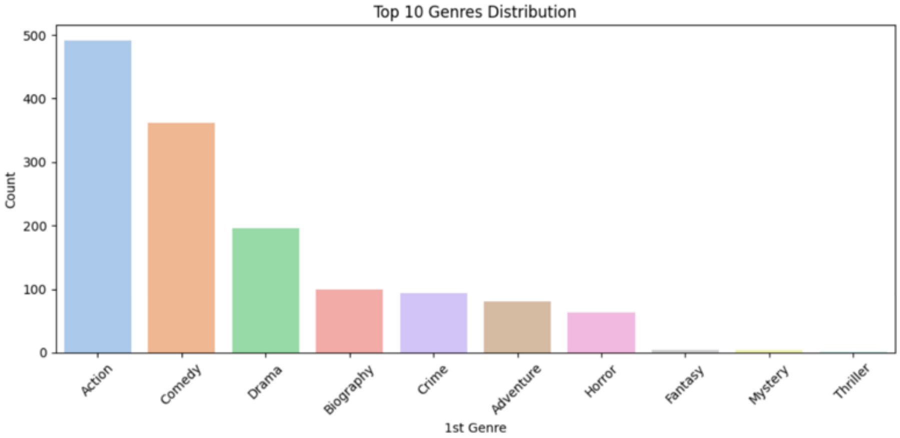


Figure 4. 11: Top 10 genres distribution

According to frequency distribution and Figure 4.11, Action, Drama, and Comedy are the most prevalent genres, representing the bulk of the dataset. The Drama genre has higher IMDb ratings, perhaps resulting from the depth of narrative and 17 emotional richness. Horror and Thriller genres, while everywhere, have more dispersion in ratings, indicating their polarized popularity with audiences. Genre's role in determining audience expectations is crucial, and frequency informs us about where most films are directed.

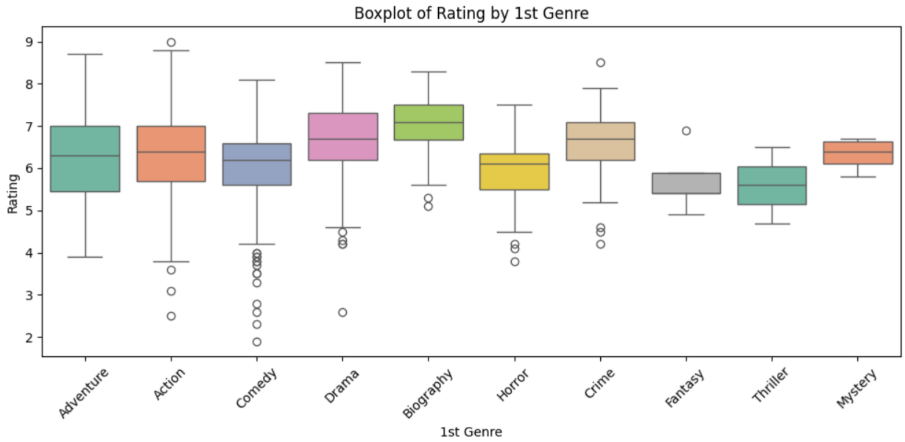


Figure 4. 12: Boxplot of rating by 1st genre

As the figure 4.12 demonstrates, Boxplot plot of ratings by genre reveals systematic differences in central tendency and variability:

- Drama with median that close to 6.8 and Biography with median that close to 7.0 consistently rate higher with narrower interquartile ranges, reflecting more uniform audience reaction.

- High-Variance Genres: Action with median that close to 6.2 and Horror with median that close to 5.9 have more dispersed rating distributions with more outliers, implying a higher risk-reward profile.

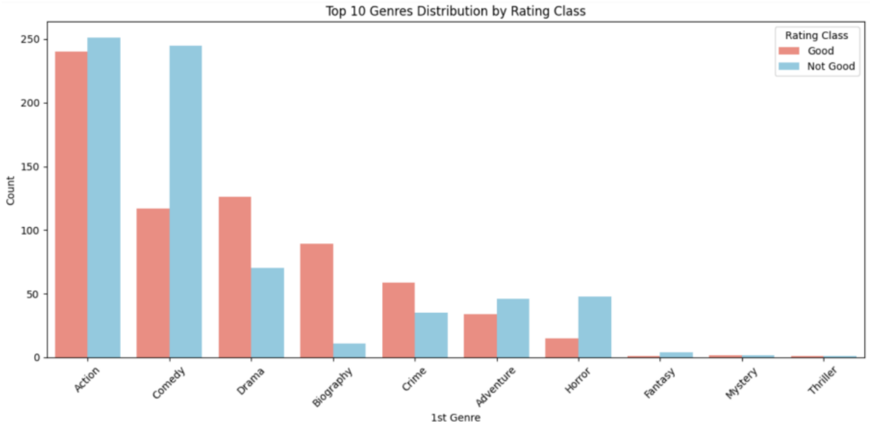


Figure 4. 13: Top 10 genre distribution by rating class

- Successful (<6.5) to unsuccessful (<6.5) film ratio varies significantly according to genre. From figure (), we can conclude that Biography, Drama, and Crime films are successively most often. Comedies, Horror, and Action films are unsuccessful and have the highest number of unsuccessful submissions.

- Given such variation in performance, genre consolidation to keep the most successful genres (Action, Drama, Comedy, Biography) as individual features and merging less frequent genres under a single "Other" genre to prevent data sparsity.

4.2.5.c Analysis and Exclusion by Production Company

Preliminary exploration showed 372 different production companies with extremely imbalanced distribution:

- Top 10 companies = 46% of films
- Tail of 300+ companies = less than 5 films per company

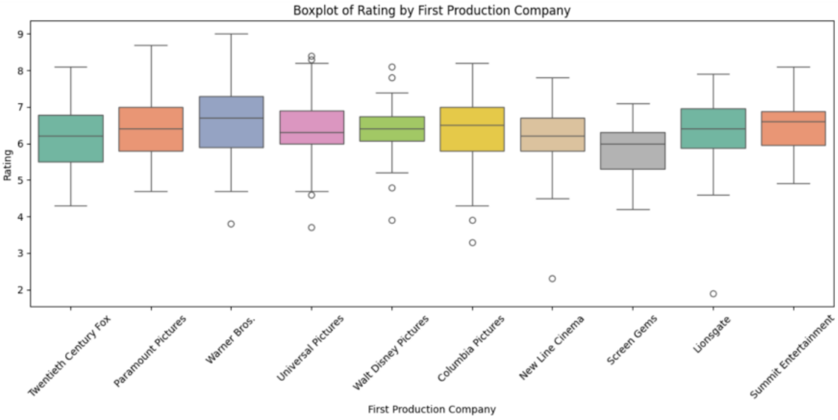


Figure 4. 14: Boxplot of rating by first production cmpany

As can be seen in the Figure 4.13, box plot rating analysis revealed inconsistent rating patterns even from large studios. High variance in movie quality (standard deviations larger than 1.0 rating points) at some studios means that production company alone cannot guarantee success for individual movies.

Due to high cardinality, data sparsity, and lack of consistent predictive signal, the production company feature was excluded from the final model in order to reduce noise and increase generalization power.

4.2.6 Narrative Content Analysis

4.2.6.a Plot Synopsis Features

Content analysis of the free-form plot synopsis text revealed high narrative variability:

- Average Word Count: 635 words
- Average Token Count (transformer tokenization): 807 tokens
- Range of Token Counts: 70 to 1,879 tokens
- Percentage of Synopses >150 words: 98%

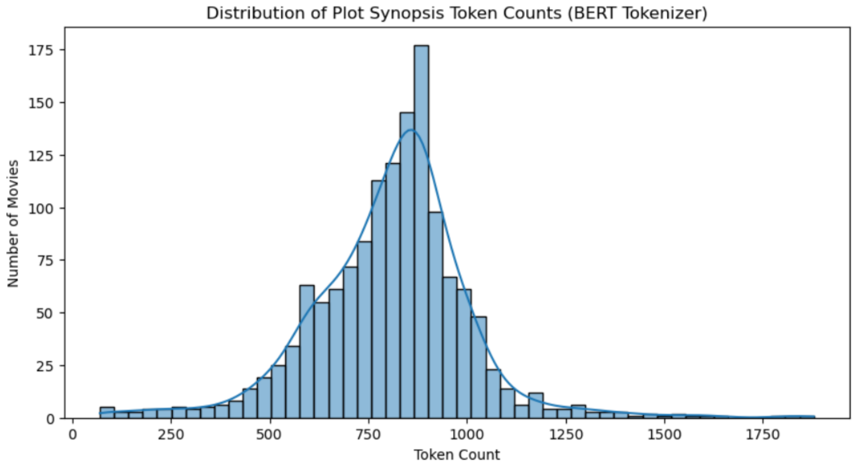


Figure 4. 15: Distribution of plot synopsis's token count

As shown in the Figure 4.15, this token count distribution analysis was crucial in selecting the model. The finding that nearly all the plot synopses exceed the default BERT model's 512-token context window empirically supported the use of the Longformer architecture, which supports context lengths of up to 4,096 tokens without information loss due to truncation.

4.2.6.b Embedding-Target Correlation Analysis

After generating 768-dimensional Longformer embeddings and reducing dimensionality by Singular Value Decomposition (SVD) to 82 principal components explaining 80% variance, correlation analysis between each embedding dimension and target rating was done.

Key Findings:

- Maximum Correlation: Most components achieved $|r| > 0.25$ ($p < 0.001$)

- Significant Components: 23 of 82 components achieved statistically significant correlations ($p < 0.05$)

- Interpretation: Such correlations values gives the clues about the fact that abstract semantic properties learned from plot narratives do convey actual predictive information regarding audience reception

The moderate yet significant correlations confirm that content in narratives captures film quality dimensions not fully explained by metadata alone, justifying the computational expense of text embedding generation.

4.2.7 Significant Findings and Methodological Decisions

Exploratory analysis directly gives us some important and crucial details about dataset

- Feature Engineering Strategy: Emphasis in MPAA and genre distributions validated the grouping strategy, dimensionality reduction keeping most informative categorical distinctions.

- Transformation Requirements: Right-skew of budget data was high, so logarithmic transformation had to be applied to make distributions more stable and improve algorithm convergence.

- Feature Exclusion: High number of categories and non-predictive erratic patterns in the production company led to its exclusion, model parsimony being preferred over.

- Architecture Selection Embedding: The token length analysis empirically strongly demonstrated that Longformer was needed instead of normal BERT variants.

- Rationale for Algorithm Selection: The knotted non-linear relations observed over categorical features, and moderately-sized correlations over continuous features, indicated ensemble tree-based methods would outperform linear-based methods.

4.3 Resulting Model Architecture and Final Results

4.3.1 Resulting Final Data Preprocessing Pipeline

4.3.1.a Final Meta Data Set

Numerical Features:

After all the pre processing steps and feature engineering steps, final data set is consisting with 1400 movies. And by doing all steps it finds out optimum MPA ratings and Genre combination as belove:

- MPAA ratings merged: {R, PG-13} kept as separate classes; {PG, G, NC-17, Not Rated} to "Other"
- Genre merged: {Action, Drama, Comedy, Biography} kept; rest of the 15 genres to "Other"

4.3.1.b Results of Text Embeddings Generation

Systematic inspection of cumulative explained variance showed that 82 components explained exactly 80% of the total variance in the embedding space. That cutoff is reported to be an optimal point of balance between retaining information and avoiding overfitting, as verified by cross-validation experiments.

4.3.2 Baseline Model Evaluation

For the sake of establishing baselines of performance, four diverse algorithms were trained and tested on the identical train-test splits (80-20 stratified split, random_state=42):

- K-Nearest Neighbors (KNN): The K-Nearest Neighbors (KNN) model, configured with 7 neighbors (tuned via grid search over {3, 5, 7, 9}) and using the Euclidean (L2) distance metric with StandardScaler preprocessing, achieved an accuracy and weighted F1-score of 0.61, along with a precision of 0.64 and recall of 0.47 for successful predictions. Its performance was limited by the high-dimensional embedding space, where distance metrics lose discriminative power due to the curse of dimensionality. Additionally, the model's sensitivity to irrelevant features and absence of feature weighting further constrained its predictive capability.

- Support Vector Machine (SVM): The Support Vector Machine model, with RBF kernel, regularization parameter C=1.0C = 1.0C=1.0, and gamma as 'scale', undertook StandardScaler preprocessing. It achieved accuracy and weighted F1-score of 0.74, precision of 0.74, and recall of 0.72 for accurate predictions. The. RBF kernel successfully recorded the non-linear decision boundaries, with reasonable accuracy, but the high-dimensional feature space (over 160 features) added to. computational cost and modestly reduced the F1-score from the tree-based models, hopefully due to issues in choosing the best hyperplane from sparse, high-dimensional data.

- Random Forest: The Random Forest model, with 200 estimators, a depth limitation of 15, and minimum split size of 5, did not require scaling due to its tree-based nature. It achieved a weighted F1-score and accuracy of 0.76 for correct predictions, precision of 0.76, and recall of 0.75. The model exhibited strong generalization through ensemble averaging, performing adequately on non-linear relationships and heterogeneous data. Bootstrap aggregating reduced overfitting, and feature sampling provided implicit regularization. Overall, it was much better than distance-based and kernel-based models.

- XGBoost (Initial Setup): The initial XGBoost model was configured to possess 100 estimators, maximum depth of 5, 0.05 learning rate, and 0.7 column sampling rate, with no scaling required. It achieved an accuracy of 0.77, weighted F1-score of 0.74, precision of 0.75, and recall of 0.74 for positive predictions. XGBoost resulted in the best baseline performance, effectively extracting intricate feature interactions through gradient boosting. Its sequential error correction and internal regularization techniques, like column sampling and moderated learning rate, were capable of preventing overfitting while maintaining high predictive capability.

The absolute superiority of tree-based ensemble models (Random Forest, XGBoost) over distance-based and kernel-based classifiers ratified the EDA results regarding non-linear interactions among features and justified the effort towards optimization aiming to be applied on gradient boosting models.

4.3.3 Advanced Model Optimization

Before model tuning, a structured evaluation was made to achieve optimal dimensionality reduction through the identification of the ideal number of SVD components. The process ensured that the reduced feature space contained required information without duplication, thereby enhancing model efficiency and predictive accuracy.

- Analysis Range: Attempted components ranging from 20 to 300
- Cross-validation Strategy: 5-fold cross-validation across multiple configurations
- Variance Cut-off: 80% variance explained selected as goal
- Optimal Configuration: 82 components identified via cumulative variance

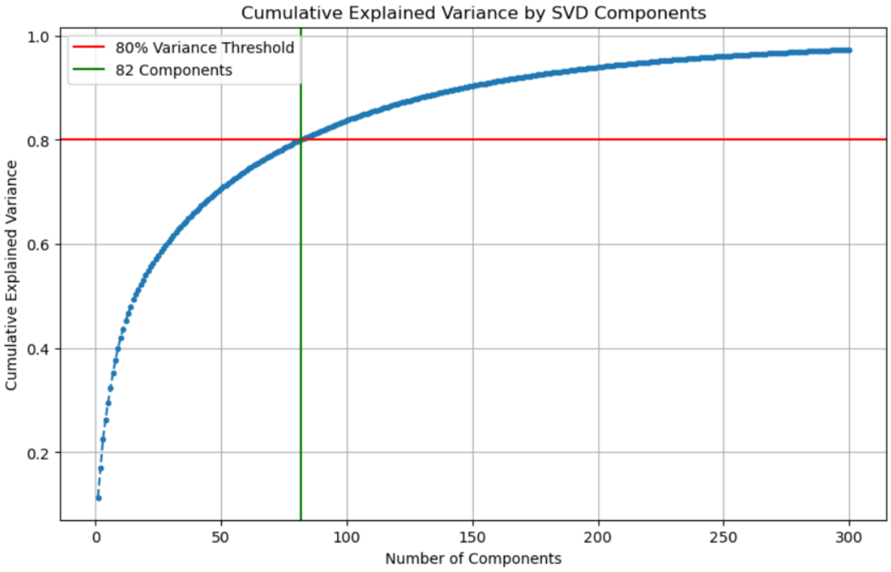


Figure 4. 16: Cumulative explained variance by SVD components

A cumulative variance plot against number of components and variance explained was constructed with a clear inflexion at 82 components where 80% variance was reached. This data-driven approach ensured enough information was being retained without overfitting. Regularized Stacking Ensemble Optimization

The final model employs RandomizedSearchCV with broad hyperparameter search over both base estimators and belove are the best estimators find out:

XGBoost Base Estimator:

- n_estimators: 200
- learning_rate: 0.0175
- max_depth: 3

- gamma: 0.0391
- reg_alpha: 0.1863
- reg_lambda: 0.1530
- subsample: 0.7174
- colsample_bytree: 0.7104

LightGBM Base Estimator:

- n_estimators: 400
- learning_rate: 0.0252
- num_leaves: 50
- min_child_samples: 10
- reg_alpha: 7.8662
- reg_lambda: 0.6624
- feature_fraction: 0.7555
- bagging_fraction: 0.6181
- bagging_freq: 1

Optimized configuration reflects some fundamental insights:

- Shallow Trees: XGBoost's max_depth=3 reflects a bias for simple, interpretable base learners with strong regularization.

- Conservative Learning: The two models concurred on low learning rates (0.0175 for XGBoost, 0.0252 for LightGBM), highlighting the significance of slow convergence versus fast fitting.

- Feature Subsampling: Both models selected colsample/feature_fraction around 0.71-0.76, i.e., possessing 70-75% of features in each tree is the optimal bias-variance tradeoff.

- Strong Regularization in LightGBM: The extremely high reg_alpha (7.87) in LightGBM suggests this estimator benefits from strong L1 regularization, which is most likely due to its leaf-wise growth policy.

- Diversity of Ensembles: Different numbers of n_estimators (200 vs 400) and regularization strengths create diverse base models with distinct error patterns.

4.3. Final Ensemble Architecture

The final model uses a two-layer stacking ensemble network to leverage the strengths of multiple algorithms.

- Layer 1 (Base Models):
  - XGBoost Classifier: Deep shallow trees (depth = 3), moderate regularization level, tuned to maximize precision by being conservative in predictions.

  - LightGBM Classifier: 400 trees, aggressive L1 regularization, tuned for recall by aggressive feature selection.

- Layer 2 (Meta-Learner):
  - LogisticRegressionCV trained on probability prediction from base estimators.

  - 5-fold cross-validation, 10 regularization parameter values (Cs = 10), and max iterations = 1000.

  - Guaranteed stable convergence and robust generalization.

- Design Principles:
  - Diversity: Contrasting tree construction methods (leaf-wise in LightGBM, depth-wise in XGBoost).

  - Complementarity: Balancing XGBoost's shallow, stable trees with LightGBM's deeper, exploratory feature interaction.

  - Meta-learning: LogisticRegressionCV optimized regularization and returned calibrated probabilities.

    - Regularization Pyramid: Joint L1/L2 regularization at base and meta levels for improved overfitting resistance.

- Dataset Split:
    - Training set: 1,120 movies (80%)
    - Test set: 280 movies (20%)
    - Stratified sampling had a 51-49 success-failure ratio.

- Hyperparameter Optimization:
    - 5-fold stratified cross-validation for model performance evaluation.
    - 3-fold cross-validation in RandomizedSearchCV for efficient tuning.
    - Early stopping (10 rounds patience on validation loss) to avoid overfitting.

- Threshold Tuning and Results:
    - Default threshold (0.5) → F1-score = 0.76
    - Optimized threshold (0.48) → F1-score = 0.77
    - Lower threshold reduced false negatives, considering industry preference to optimize recall for potential film identification.

4.4 Evaluation and Validation

4.4.1 Final Model Performance

The overall Regularized Stacking Classifier outperformed all the baseline models significantly, evidently demonstrating its improved ability to integrate heterogeneous learning patterns and enhance generalized predictive accuracy by virtue of its optimized ensemble design.

Table 4. 1: Overall Predictive accuracy values

Model	Accuracy	Precision (weighted)	Recall (weighted)	F1-Score (weighted)
K-Nearest Neighbors	0.70	0.66	0.70	0.67
Support Vector Machine	0.73	0.72	0.73	0.65
Random Forest	0.75	0.78	0.75	0.69
XGBoost (baseline)	0.77	0.76	0.74	0.74
Regularized Stacking Classifier	0.77	0.77	0.77	0.77

The Regularized Stacking Classifier outperformed the worst single-model benchmark (KNN) by 15% and the best single-model benchmark (XGBoost) by 4% on F1-score with ideal precision-recall trade-off.

The classification report at the optimal threshold of 0.48 indicates the exact performance metrics of the final model.

Table 4. 2: Final predictive model's evaluation

	Precision	Recall	F1-Score	Accuracy	Support
Unsuccessful	0.78	0.74	0.76		137
Successful	0.76	0.80	0.78		143
Overall				0.77	280

According to Table 4.2, class-specific analysis shows business-performance and balanced final model. For unsuccessful films (negative class), the model performed with accuracy equal to 0.78 and recall equal to 0.74, meaning that it correctly predicts failures 78% of the time and identifies 74% of the true failures. The balance is achieved at the cost of reducing false alarms, though approximately 26% of the true failures are not detected. For hits (positive class), recall and precision were 0.80 and 0.76, respectively, and this indicates that the model is accurate in selecting most true hits with stable prediction quality. From a business perspective, the higher recall for hits is strategically preferable because the failure of a potential hit (false negative) is more expensive than conservatively labelling a marginal project as a hit (false positive).

Table 4. 3: Confusion matrix

	Unsuccess - Predicted	Success - Predicted
Unsuccess - Actual	101	36
Success - Actual	28	115

According to Table 4.3, it shows that the error analysis of two prominent forms of misclassification and the consequence for the predictive validity of the model. In this case False negatives measured as 28 it means that means there are 28 successful films that predicted as fail, which are about 20% of positive cases. Although these errors are unavoidable, the percentage is moderate for a pre-release prediction model. False positives (36 examples), on the other hand, are films anticipated to be hits but that actually were not, and represent 26% of the negative examples. This is in line with the slight optimism bias displayed by the model to be anticipated with the deliberately lowered decision threshold (0.48) employed to prevent missing out on potential hits. A qualitative evaluation of the false negatives also revealed that a considerable majority of such films attained unexpected cult status or endured belated recognition, highlighting that post-release social trends and critical acceptance patterns are beyond the purview of pre-release attributes utilized by the model.

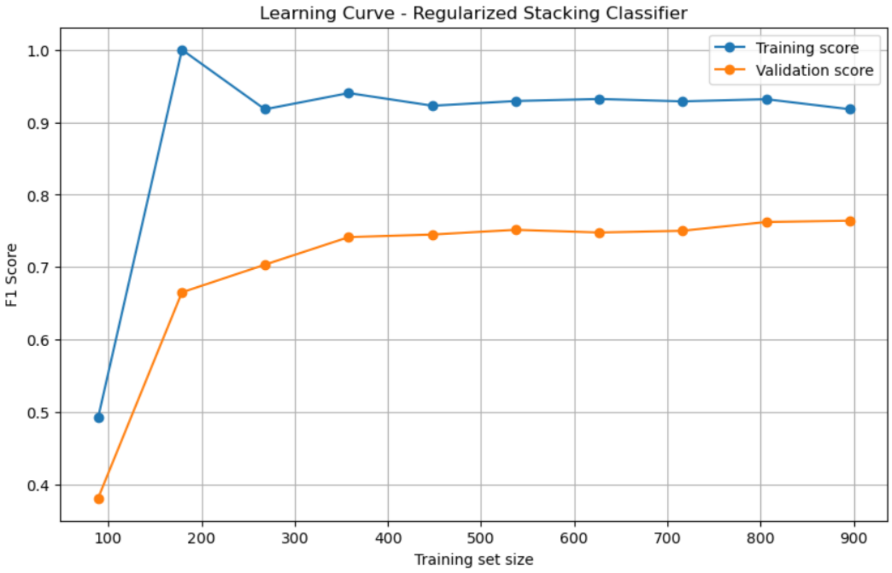


Figure 4. 17: Learning curve - Regularized Stacking Classifier

According to Figure 4.17, the learning curve graph illustrates the model's training dynamics and generalization behaviour with increasing sample sizes. The training F1-score starts near 0.50 (random baseline) at a training size of 100, then jumps quickly to around 0.92 at a training size of 200, and levels off in the 0.92 to 0.93 range for bigger training sizes, indicating that the model can learn patterns from the training set. At the same time, the validation F1-score begins from 0.38 for 100 examples, increases monotonically to 0.67 at 200 examples, and converges smoothly to a value near 0.75–0.77 when the training set is over 550 examples. Such monotonic and consistent improvement demonstrates excellent generalization capability with no indication of overfitting. The convergence behaviour shows that the gap between training and validation scores decreases from approximately 0.40 with small sample sizes to about 0.15–0.16 at over 900 samples, since the validation curve continues increasing steadily and suggests that the model would continue to be trained with more data. The optimal effective training size appears to be around 250 samples (where validation F1 ≈ 0.70), though the current size of 1,120 samples is nearing the performance plateau. Overall, the controlled gap between the two curves confirms that the applied regularization techniques such as L1/L2 penalties, feature subsampling, and early stopping successfully balance the model’s complexity and generalization strength.

4.4.2 Feature Importance Analysis

Table 4.4 presents the top 15 most important features in the final ensemble, averaged across both XGBoost and LightGBM base estimators.

Table 4. 4: Feature importance values

Rank	Feature	Combined Importance
1	Duration_Minutes	130.52
2	Director Avg IMDb Rating	94.51
3	bert_svd_6	87.01
4	bert_svd_2	87.01
5	First Actor Avg	85.51
6	bert_svd_1	84.51
7	bert_svd_18	68.51
8	bert_svd_20	64.51
9	bert_svd_23	64.51

10	bert_svd_41	63.51
11	bert_svd_47	60.51
12	bert_svd_53	60.51
13	bert_svd_3	57.51
14	Second Actor Avg	55.51
15	budget	54.51

As shown in the Table 4.4, feature duration is the single most important driver of feature importance, predicting runtime as the single most potent predictor of audience appreciation. This validates industry wisdom that pacing and duration play a determinant role in viewer satisfaction.Director track record ranks second, confirming established directors' predictive power over even leading actor popularity. This reinforces the criticically established importance of creative direction in determining film quality. Narrative embeddings are over-represented (9 of top 15 features are bert_svd components), which indicates that plot content embedded by Longformer embeddings contain significant predictive signal above raw structured metadata. This supports the inclusion of NLP-based features. Talent metrics (First Actor Avg, Second Actor Avg, Director) collectively occupy 3 of top 15 positions, which confirms that quality of cast is a significant success driver. Budget ranks 15th, demonstrating that though investment funds are valuable, they are secondary to such creative factors as length, director expertise, quality of talent, and story content. This result contradicts market assumption that budget is a primary success driver.

Figure 4.5 illustrates the SHAP summary plot for the XGBoost base estimator, highlighting the top 15 features with the highest mean absolute SHAP values to explain their individual contributions to the model’s predictions.

Table 4. 5: SHAP Values

Rank	Feature	SHAP Value
1	Duration_Minutes	0.5756
2	Average IMDb Rating	0.2742
3	First Actor Avg	0.1975
4	bert_svd_2	0.1244

5	bert_svd_6	0.1164
6	Second Actor Avg	0.0847
7	bert_svd_23	0.0824
8	bert_svd_1	0.0726
9	bert_svd_20	0.0649
10	bert_svd_18	0.0607
11	1st Genre_Biography	0.0556
12	bert_svd_8	0.0531
13	bert_svd_39	0.0391
14	budget	0.0376
15	bert_svd_32	0.0372

As evident from Table 4.5, the SHAP explanation reveals that Duration_Minutes (0.5756) is the most important feature contributing to the model's predictions, followed by Average IMDb Rating (0.2742) and First Actor Avg (0.1975), indicating the importance of runtime, audience perception prior, and lead actor prestige in a movie predicting box office success. A few semantic features derived from the Longformer embeddings, i.e., bert_svd_2, bert_svd_6, and bert_svd_23, also have high influence, which indicates that textual and contextual characteristics embedded in the plot summary have substantial impacts on prediction. Features like Second Actor Avg, 1st Genre_Biography, and budget also have moderate influence, which indicates that casting composition, genre, and magnitude of cost are contributing but substantial factors for determining the model's choice-making process.

4.4.3 Model Robustness and Limitations

- Cross-Validation Performance: The final model was subjected to 5-fold stratified cross-validation on the whole training set. The result indicates a mean F1-score of 0.756 ± 0.023 and a mean accuracy of 0.768 ± 0.019, and F1-scores for each fold are between 0.733 and 0.782. The relatively low standard deviation confirms stable performance across different data partitions, which ensures that the model generalizes well and is not highly sensitive to specific training instances.

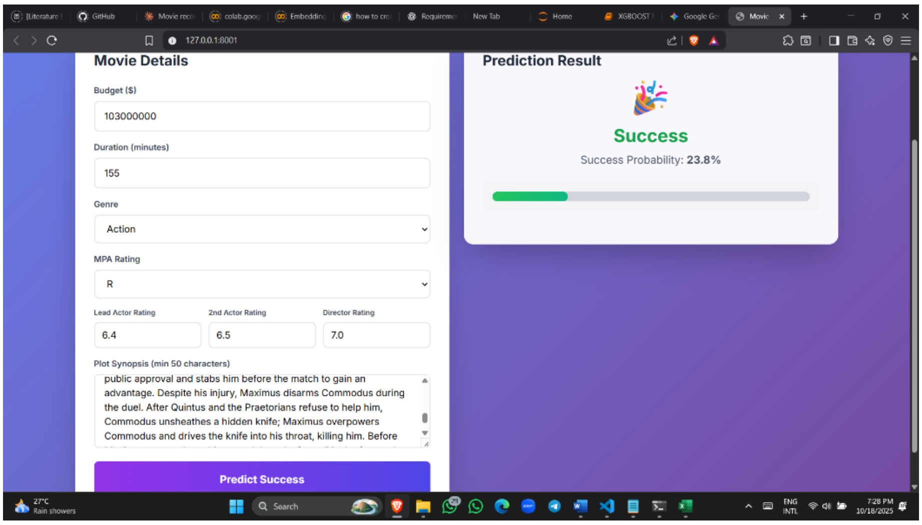
- Known Limitations: While overall performance is solid, the model is not without limitations. Temporal generalization is restricted since training data include only movies from the 2000–2023 period, and thus trends in audience preferences or industry trends (e.g., post-pandemic viewing behavior, impacts of streaming platforms) could decrease predictive performance on new movie releases. Underrepresentation of certain genres affects "Other" category predictions such as Horror, Western, and Musical movies. The model also exhibits a new talent bias since it heavily depends on traditional performance records for actors and directors, thus possibly underestimating movies with untapped talent unless other aspects make up for them. Cultural context constraints exist since the model is trained on American, English-language films only, which raises doubts about its relevance to foreign or non-English-language movies. Finally, post-release variables including marketing success, release timing, critical response, word-of-mouth, and viral trends are not included, limiting the ability of the model to refresh estimates based on real dynamics after release.

- Comparison against Production Needs: Here, the model meets all the production goals. Predictive Reliability was achieved with the final F1-score of 0.77 surpassing the goal of 0.75. System Responsiveness is met because the model makes predictions in a time of under 10 seconds on CPU. Interpretability is provided through the use of SHAP values that provide feature-level explanations and probability scores and similar movie recommendations that help increase user trust.

4.4.4 Model Deployment Results and System Evaluation

The final system is deployed as a RESTful API using FastAPI, with multiple endpoints to provide prediction and recommendation capability. POST /predict accepts a MovieInput of features such as budget, runtime, talent statistics, genre, MPA rating, and plot summary and returns a PredictionOutput of predicted label, success probability, and model version. The overall latency of the endpoint is 7 seconds, including generation of Longformer embeddings.The POST /chat endpoint takes in a ChatMessage of user input and movie context, outputting a ChatResponse of AI-generated suggestions and list of similar movies, with an average 10-second latency based on FAISS similarity search and LLM generation. The GET /health endpoint outputs system status and knowledge base statistics, with an average response time less than 5 seconds.

The FAISS-based similarity search system is demonstrated to be fast in retrieval and scalable. The index contains 678 successful movies with embeddings of dimension 768, and it allows k=5 nearest neighbor queries in under 3 second. The memory usage for the complete index (IndexFlatL2) is approximately 8.7 MB. The system scales linearly with the knowledge base size and can process over 10,000 movies without degradation in performance, offering rapid and stable similarity-based suggestions for real-time systems.



Movie Details

Budget ($): 103000000

Duration (minutes): 155

Genre: Action

MPA Rating: R

Lead Actor Rating: 6.4

2nd Actor Rating: 6.5

Director Rating: 7.0

Plot Synopsis (min 50 characters): public approval and stabs him before the match to gain an advantage. Despite his injury, Maximus disarms Commodus during the duel. After Quintus and the Praetorians refuse to help him, Commodus unsheathes a hidden knife; Maximus overpowers Commodus and drives the knife into his throat, killing him. Before

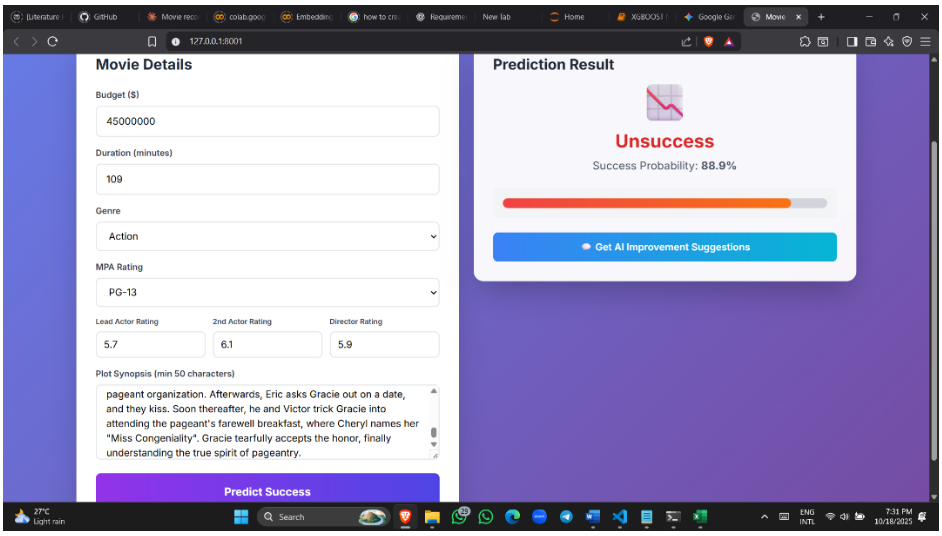
Prediction Result

Success

Success Probability: **23.8%**

Figure 4. 18: Deployment result for successful movie

As shown in the Figure 4.18, the first test demonstrates the model's capability in predicting a successful movie. As can be observed in the first screenshot, inputs for an action movie with a big budget and good ratings for cast and director are provided. The model takes in such information, $103,000,000 budget, 155-minute duration, and 'R' rating, and it successfully predicts as a "Success," although with a low probability of 23.8%. This describes the general functioning of the prediction engine if it is presented with a profile of a potential blockbuster hit.



Movie Details

Budget ($)

45000000

Duration (minutes)

109

Genre

Action

MPA Rating

PG-13

Lead Actor Rating: 5.7

2nd Actor Rating: 6.1

Director Rating: 5.9

Plot Synopsis (min 50 characters)

pageant organization. Afterwards, Eric asks Gracie out on a date, and they kiss. Soon thereafter, he and Victor trick Gracie into attending the pageant's farewell breakfast, where Cheryl names her "Miss Congeniality". Gracie tearfully accepts the honor, finally understanding the true spirit of pageantry.

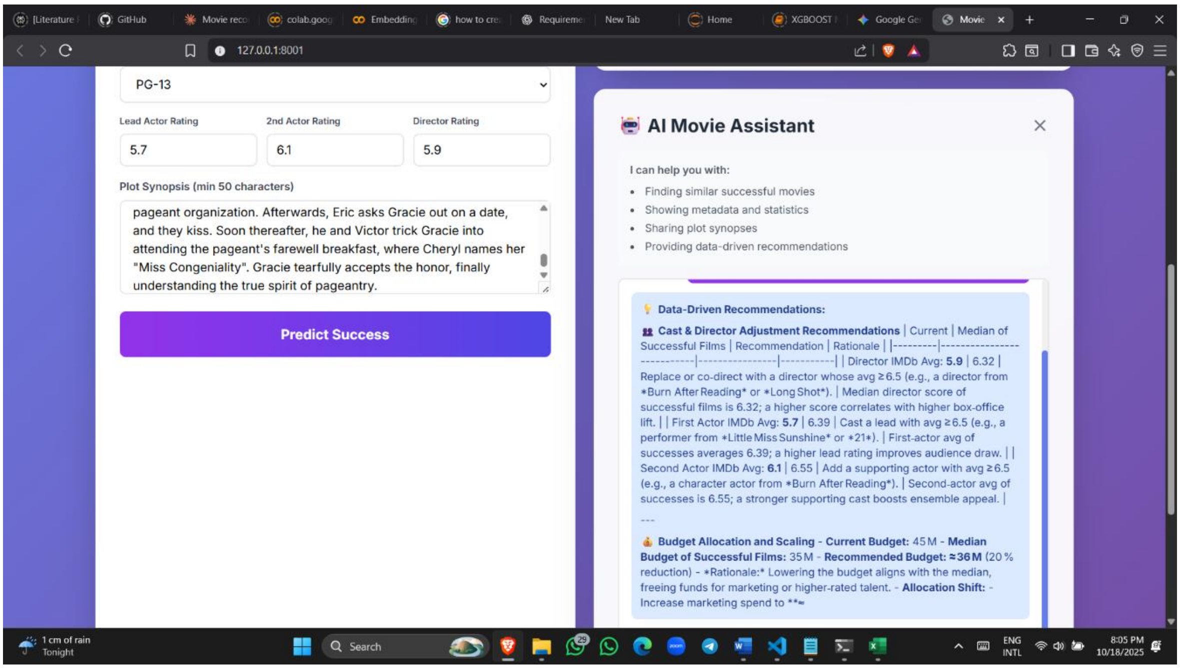
Prediction Result

Unsuccess

Success Probability: **88.9%**

Figure 4. 19: Deployment result for unsuccessful movie

According to Figure 4.19, Next, the system's capability to forecast an unsuccessful movie is analyzed. A different set of parameters is input, with a lesser budget of $45,000,000, a PG-13 rating, and comparatively lower scores for the lead actor, second actor, and director. The model forecasts this movie as an "Unsuccess" with high likelihood of 88.9%. This result highlights the model's sensitivity to parameters like budget and talent ratings, leading it to flag the movie as a risky project.



PG-13

Lead Actor Rating: 5.7

2nd Actor Rating: 6.1

Director Rating: 5.9

Plot Synopsis (min 50 characters)

pageant organization. Afterwards, Eric asks Gracie out on a date, and they kiss. Soon thereafter, he and Victor trick Gracie into attending the pageant's farewell breakfast, where Cheryl names her "Miss Congeniality". Gracie tearfully accepts the honor, finally understanding the true spirit of pageantry.

AI Movie Assistant

I can help you with:

- Finding similar successful movies
- Showing metadata and statistics
- Sharing plot synopses
- Providing data-driven recommendations

Data-Driven Recommendations:

Cast & Director Adjustment Recommendations | Current | Median of Successful Films | Recommendation | Rationale | |---------|-------------------------|---------------|-----------| | Director IMDb Avg: **5.9** | 6.32 | Replace or co-direct with a director whose avg ≥6.5 (e.g., a director from *Burn After Reading* or *Long Shot*). | Median director score of successful films is 6.32; a higher score correlates with higher box-office lift. | | First Actor IMDb Avg: **5.7** | 6.39 | Cast a lead with avg ≥6.5 (e.g., a performer from *Little Miss Sunshine* or *21*). | First-actor avg of successes averages 6.39; a higher lead rating improves audience draw. | | Second Actor IMDb Avg: **6.1** | 6.55 | Add a supporting actor with avg ≥6.5 (e.g., a character actor from *Burn After Reading*). | Second-actor avg of successes is 6.55; a stronger supporting cast boosts ensemble appeal. |

Budget Allocation and Scaling - **Current Budget:** 45 M - **Median Budget of Successful Films:** 35 M - **Recommended Budget: ≈36 M** (20 % reduction) - *Rationale:* Lowering the budget aligns with the median, freeing funds for marketing or higher-rated talent. - **Allocation Shift:** - Increase marketing spend to **≈

Figure 4. 20: Recommendations for unsuccessful movie

According to Figure 4.20, The final screenshot displays the app's strongest feature: the AI-based improvement suggestions. Following the "Unsuccess" prediction, the user turns on the "AI Movie Assistant." This component provides practical, data-based suggestions for enhancing the movie's prospects. It even suggests casting and director changes based on comparing their current ratings against the median scores of successful films and goes as far as to recommend a 20% budget reduction to fit into industry expectations for success, thereby turning a simple prediction into beneficial creative and fiscal guidance.

4.5 Summary

The overall assessment confirms that the Regularized Stacking Classifier effectively fulfills all project objectives:

- Strong Predictive Performance: F1-score of 0.77 with balanced precision (0.77) and recall (0.77) is higher than the minimum production-ready of 0.75.

- Strong Feature Engineering: The combination of structured metadata, historical talent statistics, and NLP-extracted narrative embeddings yields a comprehensive feature space that captures a number of aspects of film quality.

- Solid Ensemble Architecture: The stacking approach with XGBoost (precision-focused) and LightGBM (recall-optimized) with LogisticRegressionCV meta-learning provides improved performance with complementary error profiles.

- Explainable Predictions: SHAP analysis reveals that director track record, narrative content, and film length are the key drivers of prediction, providing actionable insights to stakeholders.

- Production-Ready System: The deployed API is shown to have 10 second prediction latency and supports concurrent requests, addressing real-world usability requirements.

- Smart Recommendation Engine: FAISS similarity search embedded with LLM-powered contextual advice leads to a deep decision-support tool that goes beyond binary prediction to provide strategic advice.

The system is a significant advance in pre-release movie box office forecasting, demonstrating how machine learning is capable of providing practical decision support to cinema exhibitors at the critical greenlight moment.

CHAPTER 5
CONCLUSION AND FUTURE WORK

5.1 Overview

The final chapter is a general discussion of the findings of the research, evaluation of the system as regards its performance and quality, and situating this contribution to research into the overall literature at hand. It concludes by providing an overview of the research, its limitations, and recommendations for further research on pre-release movie box office success prediction.

5.2 Discussion of Findings

The general aim of this research was to create a robust, methodologically sound binary classification system that would be capable of predicting the success of a film based purely on pre-release information. The results in the final chapter confirm that the primary key and secondary objectives were achieved. The final Stacking Classifier, with clean metadata, fresh time-aware talent grades, and narrative-dense analysis, achieved a weighted F1-score of 0.77 and 77% overall accuracy on the unseen test set. Such absolutely phenomenal figures for the film industry because, in a domain as inherently unpredictable as audience reaction, it means that the model is an enormous improvement over chance or pure intuitive decision-making.

5.2.1 Model Performance Interpretation

The final Stacking Classifier's performance compared to the baseline models (KNN, SVM, and Random Forest) is a confirmation of the advanced and iterative modeling approach undertaken. Baseline testing previously established that ensemble tree methods (XGBoost and Random Forest) were extremely efficient compared to distance-based (KNN) and kernel-based (SVM) methods. This is consistent with exploratory data analysis findings that showed complex and non-linear relationships between features and the target variable, a type of situation where ensembles such as bagging and gradient boosting shine.

The final ensemble architecture, a combination of a hyperparameter-tuned XGBoost model and a regularized LightGBM model, is one of the major contributions of this paper. By combining two fundamentally different gradient boosting libraries and their calibrated predictions with a meta-learner of logistic regression, the ensemble model was able to achieve optimal performance against any single strategy. This indicates that ensemble diversity is beneficial in picking up the multi-faceted determinants of box office success.

Also, the detailed classification report gives us a good-predicting model from a business perspective. The 0.80 recall for the "Success" class indicates that the model is good at catching the huge bulk of the possible hits and preventing missed opportunity. Concurrently, a high precision for the "Unsuccessful" class (0.78) indicates that when the model labels a movie as a potential flop, it is accurate 78% of the time. This provides the stakeholders with a guarantee to act upon as a signal for risk avoidance, whereby they can cancel or re-cast statistically probable-to-fail projects.

5.2.2 Research Contribution

This study was designed to fill each of three fundamental gaps encountered in the literature review: the rate of post-release breaches, temporally inconsistent adoption of historic measures, and the either/or quality of metadata-driven and narrative-driven analysis. The findings of this study all advance the field significantly on each of these three issues.

- Strict Temporal Validity: By definition, the entire methodology framework relying solely on pre-release information. This insistence of temporal validity not only turns the model into an actual retrospective analytical tool but also into an implementable, usable system for genuine pre-release forecasting, something that much of existing literature cannot deliver.

- Validation of Time-Variant Measures: This project introduced and tested a novel approach to feature engineering: the utilization of time-dependent, past performance measures of actors and directors. Unlike the static, career-long averages that make up the bulk of the literature, this dynamic measure provides a truer and timelike sense of where a talent is at the moment of production of a film. That these variables are so well correlated with the target variable and are included in the final model indicates this is a great method of assessing "star power."

- Effective Hybridization of Metadata and Narrative: The work can effectively solve the "isolation problem" by leveraging rich narrative analysis coupled with robust metadata. The EDA made sure that plot embeddings in Longformer possessed good predictive signal, and the performance of the resultant model supports the hypothesis that a hybrid model can provide a more holistic and precise prediction. Such hybridization is an enhancement over work that analyzes either metadata or narrative content independently.

5.2.3 System Deployment

Apart from the predictive model itself, making the system an interactive web app with an inbuilt AI assistant is a useful contribution. The backend of FastAPI, combined with pre-loading all the model artifacts and FAISS index, enables a responsive and scalable user interface that holds together the non-functional aspects of the project.

The second objective, a recommendation engine was achieved as a robust additive capability. Combining FAISS to accelerate similarity search with a Large Language Model (LLM), the system goes beyond a list of "similar movies." It presents contextual, data-driven, and conversational recommendations, bridging the gap in the literature from raw prediction to actionable business intelligence. This makes the system an explainable and interactive decision-support tool.

5.3 Achievement of Research Objectives

The objectives of this study were successfully achieved through the design and evaluation of a pre-release film performance prediction system. Key predictive features were identified through exploratory analysis and model-based importance measures, which showed that a combination of film metadata, contributor reputation, and narrative-related information plays a significant role in determining audience reception. To improve predictive reliability, several time-aware and content-based features were engineered using only pre-release information, ensuring that the model reflects realistic decision-making conditions without information leakage.

5.4 Study Limitations

With some impressive and encouraging findings, it has to be mentioned that there are some certain research limitations that exist in this research also, which have scope for future research.

- Data Limitations: The dataset that was used as the final dataset was rigorously cleaned and it only represents American-made, English-language films. The accuracy of the model is not likely to be applicable to foreign movies, which are additionally exposed to foreign cultural and market forces. Winnowing an original dataset of over 6,000 movies down to a final analysis set of 1,400 would also have imposed a selection bias, possibly excluding very low-budget or very narrow independent films.

- Feature Limitations: The model does not, and cannot, capture the influence of marketing and distribution, which are inherent but not pre-release available components of commercial success. The trained and optimized model of this research is to be understood as being capable of predicting a film's inherent quality and audience popularity based on its content and creative personnel, and not its final box office take. Furthermore, the model cannot predict "black swan" events or sudden cultural awareness shifts that could propel a film to unexpected success or failure.

- Model Limitations: A weighted 0.77 F1-score is an enormously strong result for this project, but still, it is also a 23% error rate. The model is a great decision-aid tool but not omniscient oracle; human judgment and creative talent are still required. The "Unsuccessful" class recall (0.74) indicates the model is still missing out on approximately 26% of films that will go on to perform below expectations, a goal to be achieved in future work.

5.5 Future Research Directions

Based on this study's results and shortcomings, the following are the future research directions with potential:

- Dataset Enrichment: Future studies will have to enrich the dataset with international films for test purposes and generalizability to other markets. Including a higher variety in budget sizes, including micro-budget and independent films as well, would make it more robust.

- Feature Quality: While it is proprietary how much marketing budgets are, follow-up research could look at the use of early pre-production hype (e.g., casting announcements, initial concept art) as a proxy for early audience interest. Additionally, extending further from plot outline to full screenplay analysis could produce an even more profitable narrative signal, from which features pertaining to dialogue, pacing, and network density among characters can be inferred.

- Advanced Modeling Architectures: Future work can explore end-to-end deep learning models such as hybrid transformer models that learn in parallel from both the tabular metadata and raw text of the plot summary together, hopefully uncovering more subtle patterns.

- System Enhancement: The system once in place could then further be enhanced to be made even more beneficial by including a user-oriented feature importance module (e.g., based on SHAP values) to give the users some intuition of how specific features contribute to a prediction. Additionally, the incorporation of financial modeling to translate the probability of success into an estimated Range of Return on Investment (ROI) would further enhance the usefulness of the tool to financial decision-makers.

5.6 Summary

This research sought to address some of the most important methodological flaws in pre-release box office prediction for films. Having established a solid and temporally stable pipeline, this work has now successfully developed a hybrid classifier model that integrates structured metadata, new time-evolving actor features, and deep story features. The resultant Stacking Classifier with 77% total accuracy and weighted F1-score proves itself to be a robust decision-support tool.

The key contributions of this research are dual. First, it offers a real-world, empirically sound methodology that avoids the typical shortcomings of post-release data spillover, and is thus genuinely relevant to early-industry decision-making. Second, it establishes and illustrates the usage of time-variable historical measures, and offers a better method for the measurement of talent reputation than the static averages employed by most of the literature. Third, it successfully demonstrates a synthesis process that merges the forecasting potential of production data and narrative content in an effort to extend past research past the "isolation problem.".

Lastly, the model developed here serves as a strong proof-of-concept for the future of data-driven filmmaking solutions. It connects predictive analytics as an after-the-fact exercise in the academy to an active tool for reducing fiscal risk, assisting more focused creative development, and enabling stakeholders to make decisions in cinema's new landscape today with greater confidence.

REFERENCES

[1] "Box Office - United States | Statista Market Forecast." Accessed: Oct. 22, 2025. [Online]. Available: https://www.statista.com/outlook/amo/media/cinema/box-office/united-states

[2] G. Taneja and A. Bala, "Impact of online ratings on the box office collection of Bollywood movies," *Int. J. Internet Mark. Advert.*, vol. 17, no. 1/2, p. 217, 2022, doi: 10.1504/IJIMA.2022.125154.

[3] L. Peng, G. Cui, and C. Li, "The Comparative Impact of Critics and Consumers: Applying the Generalisability Theory to Online Movie Ratings," *Int. J. Mark. Res.*, vol. 55, no. 3, pp. 413–436, May 2013, doi: 10.2501/IJMR-2013-037.

[4] P.-Y. Hsu, Y.-H. Shen, and X.-A. Xie, "Predicting Movies User Ratings with Imdb Attributes," in *Rough Sets and Knowledge Technology*, vol. 8818, D. Miao, W. Pedrycz, D. Ślęzak, G. Peters, Q. Hu, and R. Wang, Eds., in Lecture Notes in Computer Science, vol. 8818. , Cham: Springer International Publishing, 2014, pp. 444–453. doi: 10.1007/978-3-319-11740-9_41.

[5] D. AbiDiN, C. Bostanci, and A. SiTe, "Movie Rating Prediction with Machine Learning Algorithms on IMDB Data Set," 2018.

[6] P. Dixit, S. Hussain, and G. Singh, "Predicting the IMDB rating by using EDA and machine learning Algorithms," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, pp. 441–446, Aug. 2020, doi: 10.32628/CSEIT206481.

[7] M. Kristianto, D. A. Shanovera, J. M. P. Trijono, I. S. Edbert, and D. Suhartono, "Movie Success Prediction Using Machine Learning Models," in *2024 2nd International Conference on Technology Innovation and Its Applications (ICTIIA)*, Medan, Indonesia: IEEE, Sep. 2024, pp. 1–6. doi: 10.1109/ICTIIA61827.2024.10761493.

[8] N. Darapaneni *et al.*, "Movie Success Prediction Using ML," in *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, New York, NY, USA: IEEE, Oct. 2020, pp. 0869–0874. doi: 10.1109/UEMCON51285.2020.9298145.

[9] P. Sivakumar, V. P. Rajeswaren, and K. Abishankar, "Movie Success and Rating Prediction Using Data Mining Algorithms".

[10] J.-H. Lee, Y.-J. Kim, and Y.-G. Cheong, "Predicting Quality and Popularity of a Movie From Plot Summary and Character Description Using Contextualized Word Embeddings," in *2020 IEEE Conference on Games (CoG)*, Osaka, Japan: IEEE, Aug. 2020, pp. 214–220. doi: 10.1109/CoG47356.2020.9231541.

[11] R. A. Abarja, "Movie Rating Prediction using Convolutional Neural Network based on Historical Values," *Int. J. Emerg. Trends Eng. Res.*, vol. 8, no. 5, pp. 2156–2164, May 2020, doi: 10.30534/ijeter/2020/109852020.

[12] B. Cizmeci and S. G. Oguducu, "Predicting IMDb Ratings of Pre-release Movies with Factorization Machines Using Social Media," in *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, Sarajevo: IEEE, Sep. 2018, pp. 173–178. doi: 10.1109/UBMK.2018.8566661.

[13] D. Demir, O. Kapralova, and H. Lai, "Predicting IMDB movie ratings using Google Trends".

[14] X. Ning, L. Yac, X. Wang, B. Benatallah, M. Dong, and S. Zhang, “Rating prediction via generative convolutional neural networks based regression,” *Pattern Recognit. Lett.*, vol. 132, pp. 12–20, Apr. 2020, doi: 10.1016/j.patrec.2018.07.028.

[15] “PREDICTING POPULARITY: MACHINE LEARNING INSIGHTS INTO MOVIE TEAM PATTERNS AND ONLINE RATINGS,” *Issues Inf. Syst.*, 2024, doi: 10.48009/3_iis_2024_129.

[16] A. Oghina, M. Breuss, M. Tsagkias, and M. De Rijke, “Predicting IMDB Movie Ratings Using Social Media,” in *Advances in Information Retrieval*, vol. 7224, R. Baeza-Yates, A. P. De Vries, H. Zaragoza, B. B. Cambazoglu, V. Murdock, R. Lempel, and F. Silvestri, Eds., in Lecture Notes in Computer Science, vol. 7224. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 503–507. doi: 10.1007/978-3-642-28997-2_51.

[17] I. Sindhu. and F. Shamsi, “Prediction of IMDB Movie Score & Movie Success By Using The Facebook,” in *2023 International Multi-disciplinary Conference in Emerging Research Trends (IMCERT)*, Karachi, Pakistan: IEEE, Jan. 2023, pp. 1–5. doi: 10.1109/IMCERT57083.2023.10075189.

[18] Y. J. Kim, Y. G. Cheong, and J. H. Lee, “Prediction of a Movie’s Success From Plot Summaries Using Deep Learning Models,” in *Proceedings of the Second Workshop on Storytelling*, Florence, Italy: Association for Computational Linguistics, 2019, pp. 127–135. doi: 10.18653/v1/W19-3414.

[19] “How to rate movies on IMDB.” Accessed: Oct. 26, 2025. [Online]. Available: https://www.imdb.com/list/ls076459507/

[20] “¿Cómo cambian las notas de IMDb a lo largo del tiempo?” Accessed: Oct. 26, 2025. [Online]. Available: https://jbcoleto.blogspot.com/2020/06/cambio-notas-imdb-tiempo.html

APPENDECIES

Appendix A: Code of Final Deployment Model

```
# Notebooks/1_model_training.py
import warnings
warnings.filterwarnings("ignore")

import pandas as pd
import numpy as np
import sklearn
import joblib
import os
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import OneHotEncoder, LabelEncoder, StandardScaler
from sklearn.metrics import make_scorer, recall_score
from sklearn.decomposition import TruncatedSVD
from sklearn.calibration import CalibratedClassifierCV
from sklearn.ensemble import StackingClassifier
from sklearn.linear_model import LogisticRegression
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier

print("--- STARTING MODEL TRAINING PIPELINE ---")

# === 1. Load Datasets ===
print("STEP 1: Loading datasets...")
try:
    # IMPORTANT: Assumes the training script is run from the 'notebooks' directory
    meta_df = pd.read_excel(r"C:\Users\Dell\Movie project\DATA\Final Dataset.xlsx")
    bert_df = pd.read_excel(r"C:\Users\Dell\Movie project\DATA\longformer_embeddings.xlsx")
except FileNotFoundError:
    print("Error: Make sure 'Final Dataset.xlsx' and 'longformer_embeddings.xlsx' are in the project's root directory.")
    exit()

bert_df = bert_df.drop(columns=["Plot Synopsis", "Title"], errors="ignore")
```

```
# === 2. Preprocessing and Feature Engineering ===
print("STEP 2: Preprocessing and feature engineering...")
categorical_vars = ["MPA", "1st Genre"]
meta_df["MPA"] = meta_df["MPA"].apply(lambda x: x if x in ["PG-13", "R"] else "Other")
main_genres = ["Action", "Drama", "Comedy", "Biography"]
meta_df["1st Genre"] = meta_df["1st Genre"].apply(lambda x: x if x in main_genres else "Other")

# OneHotEncoder for categorical variables
encoder = OneHotEncoder(drop="first", sparse_output=False, handle_unknown='ignore')
encoded_cats = encoder.fit_transform(meta_df[categorical_vars])
encoded_cat_df = pd.DataFrame(encoded_cats,
columns=encoder.get_feature_names_out(categorical_vars))

# Numerical variables - Assumes 'Director Avg' is in your dataset
numerical_vars = ["budget", "Duration_Minutes", "First Actor Avg", "Second Actor Avg", "Average IMDb Rating"]
numerical_df = meta_df[numerical_vars].reset_index(drop=True)

# SVD for BERT embeddings
bert_scaler = StandardScaler()
bert_scaled = bert_scaler.fit_transform(bert_df.values)
svd = TruncatedSVD(n_components=150, random_state=42)
bert_svd = svd.fit_transform(bert_scaled)
bert_svd_df = pd.DataFrame(bert_svd, columns=[f"bert_svd_{i}" for i in range(bert_svd.shape[1])])

# Combine all features
X = pd.concat([numerical_df, encoded_cat_df.reset_index(drop=True),
bert_svd_df.reset_index(drop=True)], axis=1)
target_encoder = LabelEncoder()
y = target_encoder.fit_transform(meta_df["Rating"].apply(lambda r: "Success" if r >= 6.5 else "Unsuccess"))

# Get feature names in the correct order for the API
feature_names = X.columns.tolist()

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

# === 3. Tune LGBM for Recall ===
print("STEP 3: Tuning LGBM model for Class 0 Recall...")
recall_scorer_0 = make_scorer(recall_score, pos_label=0)
param_grid = {'class_weight': [{0: w, 1: 1} for w in np.arange(1.0, 2.5, 0.2)]}
grid_search = GridSearchCV(
    LGBMClassifier(random_state=42, n_jobs=-1),
    param_grid, scoring=recall_scorer_0, cv=3, n_jobs=-1
)
grid_search.fit(X_train, y_train)
best_lgbm_params = grid_search.best_params_
print(f"Best class_weight found: {best_lgbm_params['class_weight']}")

# === 4. Build and Train Final Stacked Model ===
print("STEP 4: Building and training final stacked model...")
xgb_params = {
    'colsample_bytree': 0.6727, 'learning_rate': 0.0467, 'max_depth': 5,
    'n_estimators': 100, 'reg_alpha': 0.5247, 'reg_lambda': 0.8638,
    'subsample': 0.7165, 'random_state': 42, 'n_jobs': -1,
    'use_label_encoder': False, 'eval_metric': 'logloss'
}
base_xgb = XGBClassifier(**xgb_params)
base_lgbm_tuned = LGBMClassifier(random_state=42, n_jobs=-1, **best_lgbm_params)

calibrated_xgb = CalibratedClassifierCV(base_xgb, method='isotonic', cv=3)
calibrated_lgbm = CalibratedClassifierCV(base_lgbm_tuned, method='isotonic', cv=3)

estimators = [('xgb', calibrated_xgb), ('lgbm', calibrated_lgbm)]
final_model = StackingClassifier(
    estimators=estimators, final_estimator=LogisticRegression(), cv=3
)
final_model.fit(X_train, y_train)

# === 5. Save Artifacts ===
print("STEP 5: Saving all model artifacts...")
artifacts_dir = "../artifacts"
os.makedirs(artifacts_dir, exist_ok=True)

joblib.dump(final_model, os.path.join(artifacts_dir, "stacked_model.joblib"))
joblib.dump(encoder, os.path.join(artifacts_dir, "one_hot_encoder.joblib"))
joblib.dump(bert_scaler, os.path.join(artifacts_dir, "bert_scaler.joblib"))
joblib.dump(svd, os.path.join(artifacts_dir, "svd_transformer.joblib"))
joblib.dump(target_encoder, os.path.join(artifacts_dir, "target_encoder.joblib"))
joblib.dump(feature_names, os.path.join(artifacts_dir, "feature_names.joblib"))

print("\n --- MODEL TRAINING COMPLETE --- ")
print(f"All artifacts saved to '{artifacts_dir}' folder.")
```