Predicting Movie Success at the Box Office

Ankit Bisht
ankit21014@iiitd.ac.in

Aniket Panchal
aniket21448@iiitd.ac.in

Aryan Sharma
aryan21454@iiitd.ac.in

Syam Sai Santosh Bandi
syam22528@iiitd.ac.in

Aditya Jagadale
aditya22032@iiitd.ac.in

Abstract

The goal of this research is to use multiple machine learning models to predict box office success for films. This originates from a desire to understand and analyse the aspects that influence a movie's box office success, which is classified as "flop," "average," or "hit." Such projections will assist production businesses and investors in making sound judgements based on past film data.

***Github Link**: Movie Success Prediction*

1. Introduction

As we know the box office performance of movies are not influenced by a single variable, It depends on multiple variable ,hence predicting success of a movie is a very challenging task.The above factors involve all these variables, from the budget of the movie's production, star presence, genre, marketing campaigns, release dates, and reviews from the audience. All these can play a major role in whether a film is a blockbuster or a box-office failure or whether it is just average. Predictive, approximate monetary success or failure at the box office is critical for both production companies and distributors, as well as marketing teams, since it may guide important decisions about budgeting, marketing strategies, distribution channels, and even casting.

The purpose of this project is to use machine learning techniques to categorise movies as flops, averages, or hits based on historical data and movie-related criteria. By training machine learning models on these data, the research hopes to create a reliable classification system that can help in predicting the success of new films before they are released. The capacity to classify films as probable flops, averages, or hits can give production firms with significant insights into the aspects that contribute most to a film's success.

2. Literature Survey

Quader, Nahid, Gani, Md, Chaki, Dipankar, and Ali's paper discusses a decision support system that helps investors in the movie industry avoid financial risks by predicting a movie's success based on profitability. Using machine learning techniques like Support Vector Machine (SVM), Neural Networks, and Natural Language Processing, the system analyzes historical data from sources such as IMDb, Rotten Tomatoes, Box Office Mojo, and Metacritic. The authors focus on pre-released and post-released features for the prediction, such as budget, IMDb votes, and the number of screens. They propose a model that classifies movies into five categories, ranging from flop to blockbuster, and give weight to factors like budget and star power are key indicators of success.The study achieves 84.1% accuracy with Neural Networks and 83.44% with SVM for pre-release features, with improved results when considering all features. This research adds to the growing body of work on movie success prediction by integrating both pre- and post-release data, offering investors a more comprehensive and practical tool for decision-making in the high-stakes film industry. [1].

Lee, Park, Kim, and Choi proposed the Cinema Ensemble Model (CEM), a robust machine learning-based approach to predict movie box-office success. This model enhances prediction accuracy by integrating an ensemble of algorithms, including Gradient Tree Boosting, Random Forests, and Logistic Regression. A key innovation in their work is the inclusion of transmedia storytelling—a feature based on leveraging narratives across multiple media platforms, which has been shown to drive greater audience engagement and success. By incorporating pre-release features such as marketing buzz and star power alongside post-release attributes like early box-office performance, the model improves significantly over previous studies, achieving an accuracy of 58.5 %, surpassing earlier models by Sharda and Delen (2006) and Zhang et al. (2009). However, the authors highlight the sensitivity of algorithms like Support Vector Machines to overfitting, particularly in handling pre-release data, suggesting that future research should address these limitations to further enhance prediction accuracy [2].

3. Dataset and Preprocessing

The dataset used for this project was collected from three primary sources: IMDb, TMDb, and Kaggle. Initially, the dataset consisted of 1.1 million entries, with the following attributes:

id, title, vote average, vote count, status, release date, revenue, runtime, adult, budget, imdb id, original language, original title, popularity, genres, production companies, production countries, spoken languages, keywords, directors, writers, primary director

After identifying missing data, we proceeded to clean and preprocess the dataset to ensure its suitability for machine learning models.

3.1. Preprocessing Steps

We cleaned the dataset by:

- Removing null values across key features such as budget and runtime.
- Removing non-significant columns that would not significantly impact the prediction of a movie's success like IMDb ID, original title, release date, status, and ID.
- Converting categorical features like genres and production companies into numerical representations using label encoding.
- Creating a target variable by binning the vote average into three categories: Flop, Average, and Hit.

This resulted in a dataset ready for training with machine learning models.

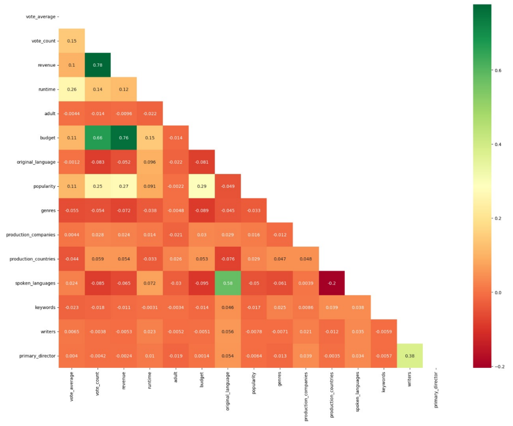

Figure 1. Correlation Heatmap of Movie Features.This heatmap should correlation between different features of our dataset. As we can see there is strong correlation between revenue and vote count, hence to remove multicollinearity we have dropped revenue column.

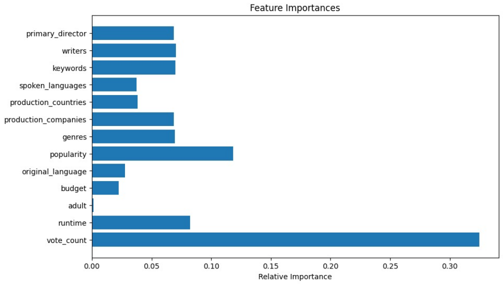

Figure 2. Relative Importance of Movie Features. This bar chart shows the relative importance of movie features in predicting movie success. Vote Count is the most important feature in predicting

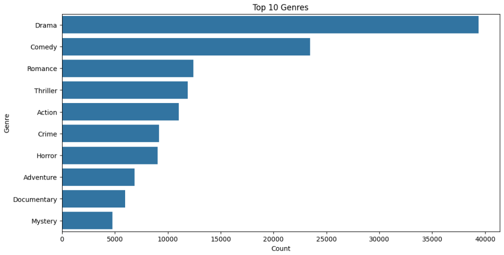

Figure 3. Top 10 movie genres in the data

4. Methodology and Models

In our effort to predict movie success, we experimented with several machine learning models, each model having different strategy to classify the movies into this classes : *Flop*, *Average*, or *Hit*. Below is a detailed explanation of the models tested and their performance:

- **Decision Tree Classifier**: The Decision Tree Classifier is a simple machine learning model that works by splitting the data into subsets based on the most important features. The model creates a tree structure where each internal node represents a feature, each branch represents a decision, and each leaf node represents predicting class. The decision trees are easy to interpret but they leads to overfitting easily especially when the tree becomes overly complex.

  Initially, the Decision Tree achieved an **accuracy of 69%**, indicating room for improvement. To address this, we applied **hyperparameter tuning** using GridSearchCV. Parameters such as `max_depth`, `min_samples_split`, and `min_samples_leaf` were optimized, resulting in the best configuration: `max_depth=15`, `min_samples_split=5`, and `min_samples_leaf=1`. The tuned model improved its accuracy to **76%**, representing a **7% increase in accuracy** compared to the initial model.

Additionally, **k-fold cross-validation** (5 folds) was used to ensure the robustness of the model. The mean accuracy across folds was **76%**, with a minimal standard deviation of 0.00, highlighting the model's consistency on unseen data.

- **Random Forest Classifier**: Random Forest is an ensemble-based machine learning model that constructs multiple decision trees during training and outputs the class that is the mode of the classifications of the individual trees.It is very effective in handling large number of features as compare decision tree classifier and dealing with non linear relationship between the features. Random Forest reduces the likelihood of overfitting by averaging the predictions of numerous trees, thus creating a more robust and generalized model.

  Initially, the model achieved an **accuracy of 78%**. To improve its performance, we applied **hyperparameter tuning** using GridSearchCV, optimizing parameters such as the number of estimators (`n_estimators`), maximum depth (`max_depth`), and minimum samples for splits and leaves. The best parameters were:

  ```
  n_estimators=150,
  max_depth=None,
  max_features='sqrt',
  min_samples_split=10,
  min_samples_leaf=1.
  ```

  With this configuration, the **testing accuracy improved to 78.5%**, and the **mean cross-validation accuracy** was 78.96%, demonstrating robust and consistent performance.

- **XGBoost Classifier**: XGBoost, or Extreme Gradient Boosting, is a highly efficient and scalable implementation of gradient boosting algorithms. It builds an ensemble of decision trees in a sequential manner, where each tree corrects the errors of the previous ones by optimizing a loss function. XGBoost incorporates advanced regularization techniques (L1 and L2) to prevent overfitting, making it particularly effective in scenarios with high-dimensional data and complex relationships between features.

  After conducting **hyperparameter tuning** using GridSearchCV, the best parameters were determined to be:

  ```
  n_estimators=100,
  max_depth=10,
  learning_rate=0.05,
  subsample=0.8,
  colsample_bytree=0.8.
  ```

  With these optimized parameters, the model achieved a **testing accuracy of 79%**, demonstrating strong performance in classifying movies. The **weighted F1 score of 0.75** reflects a good balance between precision and recall across the classes. The **mean cross-validation accuracy** was **79.02%**, indicating consistent performance across different data splits.

  The confusion matrix revealed high precision and recall for the "Average" and "Flop" classes but some difficulty in correctly identifying movies in the "Hit" category. Despite this, the XGBoost model outperformed several others, confirming its robustness and effectiveness in handling this dataset.

- **LightGBM Classifier**: LightGBM (Light Gradient Boosting Machine) is a fast, efficient, and scalable gradient boosting framework that uses histogram-based techniques for feature binning, making it well-suited for large datasets. It builds decision trees leaf-wise instead of level-wise, reducing loss more efficiently and improving accuracy while maintaining low computational costs. LightGBM is particularly effective in handling categorical features and datasets with large feature spaces.

  After hyperparameter tuning, the LightGBM classifier emerged as the best-performing model. It achieved a test accuracy of **78.71%** with a weighted F1 score of **0.754**, indicating its effectiveness in classifying movies into the respected categories. The model's ability to handle categorical features and large datasets efficiently contributed to its superior performance.

  The best parameters identified through GridSearchCV were:

  ```
  colsample_bytree: 0.8,
  learning_rate: 0.05,
  max_depth: None,
  n_estimators: 100,
  num_leaves: 50,
  subsample: 0.8
  ```

  Cross-validation confirmed the model's robustness, with scores ranging from **78.77% to 79.39%**, yielding a mean score of **79.07%**. These results highlight LightGBM's capability to generalize well across diverse datasets and its potential for accurate predictions in real-world scenarios.

5. Results and Analysis

The **LightGBM model achieved the best performance**, with an **accuracy of 79.06%.** Precision: 77% for Flops, 95% for Average, 67% for Hits. Recall: 97% for Flops, 63% for Average, 19% for Hits. F1-Score: 0.75 Balancing precision and recall. The second-best model, the xgBoost model, achieves an **accuracy of 79.02%**. Considering the fact that the lgbm model is slightly better than the xg boost and that the xg boost is relatively faster than the lgbm, we can analyse the tradeoff between speed and accuracy.

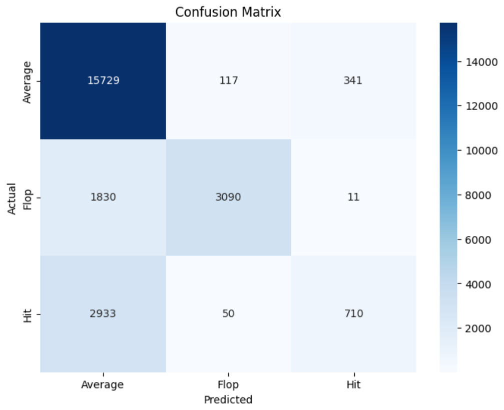

Figure 4. The diagram shows the confusion matrix of the LightGBM Classifier. Since the diagonal blocks represent the correct predictions, we can see that the model is performing well for the Average category, but not so well for the Hit and Flop categories. This can be happen because of the imbalance in the dataset, where the Average category has more instances compared to the Hit and Flop categories.

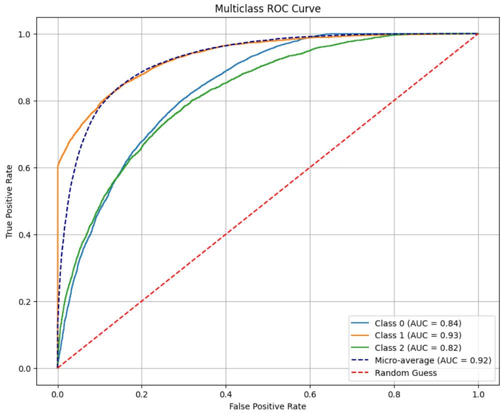

Figure 5. The ROC curve visualizes the classifier's performance for a multiclass problem. **Class 1 - AVERAGE** has the highest AUC (0.93), indicating the best separability, followed by **Class 0 - FLOP** (0.84) and **Class 2 - HIT** (0.82). The micro-average AUC (0.92) summarizes overall performance across all classes.

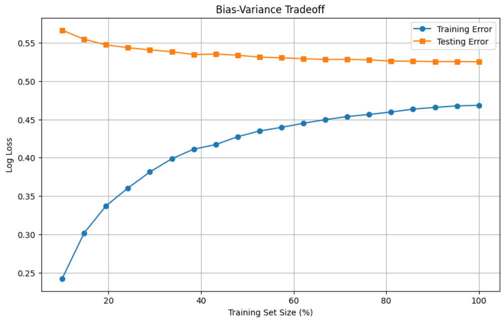

Figure 6. The graph shows the bias variance tradeoff for the LGBM classifier

As training size increases, the gap between training and testing error remains small, indicating good generalization. Log loss plateaued, prompting the use of GridSearch for hyperparameter tuning. With low training sizes (10%-20%), the model overfits, showing low training error but high testing error. As training size grows (40%-100%), the error gap narrows, improving the bias-variance tradeoff, with testing error plateauing before training error, suggesting a need for further tuning or complexity adjustment.

6. Conclusion

Throughout this project, we successfully implemented and compared multiple machine learning models, analyzing their strengths and weaknesses. We gained a strong understanding of the machine learning pipeline, including data preprocessing, feature engineering, and model evaluation, while addressing challenges such as class imbalance and metric selection. LightGBM emerged as the most effective model. Handling imbalanced data, selecting impactful features, and optimizing hyperparameters required careful analysis and trade-offs to ensure balanced performance across classes.

7. Contribution

Syam and Aditya: Data collection and Model training, Analysis

Aryan and Ankit: Exploratory Data Analysis and Model training

Aniket: Data transformation, Model training and Model Evaluation

References

[1] T. Quader, S. Nahid, M. Gani, D. Chaki, and M. Ali, ”Predicting Box Office Success: An Application of Machine Learning,” *International Journal of Computing and Digital Systems*, vol. 9, no. 2, pp. 139-147, 2020.

[2] K. Lee, S. Park, H. Kim, and J. Choi, ”Cinema Ensemble Model (CEM): Enhancing Box Office Success Prediction Using Transmedia Storytelling,” *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 4, pp. 86-94, 2020.