Film Industry Analysis

(Machine Learning Model to predict the movie success)

31.10.2023

By Group 2

- Myo Thiha,
- Rakshya Lama Moktan

Introduction

The film sector encounters a significant issue in foreseeing the success of movies, leading to substantial annual financial losses. Despite substantial investments and a distinguished cast, a film's profitability remains uncertain. This project aims to tackle this challenge by utilizing extensive historical film data to analyze and forecast critical factors impacting a film's success. The objective is to offer valuable insights into the film industry and the determinants influencing audience engagement and box office earnings.

Problem Statement

The primary challenge confronting the film industry is the inherent uncertainty surrounding a film's success. Despite substantial investments and a proficient cast, ensuring a profitable outcome is uncertain. The objective is to analyze historical film data, identify patterns, and pinpoint key attributes affecting film success. Subsequently, the aim is to utilize these identified characteristics for predicting potential film performance, encompassing audience reception and box office revenue.

Related Work

To tackle this issue, our approach will draw upon established research and relevant studies, such as "Box Office Forecasting" (BoxOfficePro), offering insights into box office dynamics, and forecasts from the "Streaming Platform" (Netflix), shedding light on entertainment industry trends. Additionally, academic works like "Forecasting Box Office Revenues" (Ruus and Sharma, 2019) and "Predicting Movie Success" (Darapaneni et al., 2020) provide valuable perspectives on predicting film success through the utilization of metadata, reviews, and machine learning algorithms.

Dataset

For this study, two datasets will be utilized:

TMDB Dataset:

Comprising metadata for 45,000 films released prior to July 2017, sourced from TMDB (Movie Database) and GroupLens. This dataset encompasses information such as cast, crew, plot keywords, budget, revenue, release date, language, production details, and user ratings.

IMDb Dataset:

Offering information on movie attributes, including IMDb Movie ID, Release Year, Certificate, Runtime performance, genre, ratings, description, director, stars, votes, and box office gross. The inclusion of additional details like director and star IMDb IDs enhances the granularity and precision of the dataset.

Methodology

Our project will follow a structured methodology that includes:

Initial Preprocessing (Before EDA)

Merging of datasets

The integration of TMDB's dispersed genre-specific movie dataset involved merging through an outer join, utilizing the common movie_id field, after adjusting for the removal of 'tt' from TMDB IDs. Subsequently, the datasets from TMDB and IMDb, initially separated, were merged using a right join, considering the prevalence of movies on both platforms and the additional information available on TMDB. To streamline the datasets, certain columns such as "video," "poster_path," "homepage," "overview," "tagline," and "description" were eliminated due to their individual movie specificity and limited computational relevance. The shape of our initial unclean merged dataset was (78116, 14)].

Merging of columns

Following the dataset merge, numerous NaN columns and redundancies needed attention. Redundancies included replicated genre columns from IMDb and TMDB, which were aligned by removing 'id' from IMDb's genre column and splitting the 'genre_y' column from TMDB. The process involved:

1. Conversion of IMDb genre from dictionary to list.
2. Merging IMDb and TMDB genre columns, retaining unique genres from both.
3. Converting the genre list back to a set.
4. Eliminating redundant genre columns.
5. Dropping additional columns, including 'movie_id,' 'tmdbId,' 'imdbId,' 'id,' and 'original_title.'
6. Extracting the year from the release_date and merging it with the existing release_year column.

7. Merging final redundant columns 'movie_title' and 'title.'

The resultant dataset, refined through these preprocessing steps, was then prepared for Exploratory Data Analysis (EDA).

Feature Extrapolation

As our merged dataset is huge combined we decided to perform feature extrapolation for cleaner and less redundant data. We started with cleaning the belongs_to_collection columns which addresses whether the movie has a prequel or not. We applied a lambda collection to get only the name of the collection the movie belongs to along with its ID in a new column for each. We then preprocessed the language columns where we extracted the 3 languages the movie was produced in different columns(only 3 is an arbitrary number for easier computing). A similar approach was also implemented for production_companies and country_data where we extracted them to different columns (only 3 were also taken here).

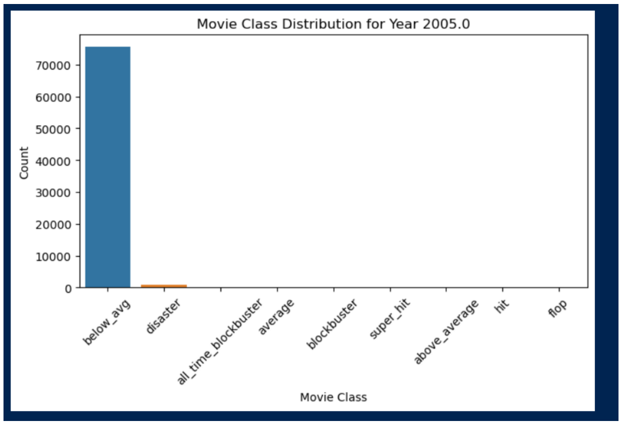
We then proceeded to extrapolate the genre columns to separate columns for each unique genre of the movie where if present it was set to 1 else to 0. For example: if a movie has humor in its genre then the value for that specific movie record is set to 0. We also combined the revenue_x and revenue_y to revenue, runtime_x and runtime_y to the runtime column using combine_first which combines two DataFrame objects and uses the value from the second DataFrame if the first has a NULL value. The final shape of the cleaned dataset was (1306179, 74).

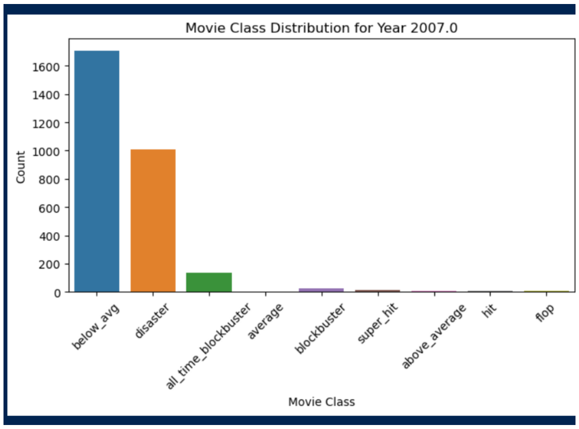
Exploratory Data Analysis

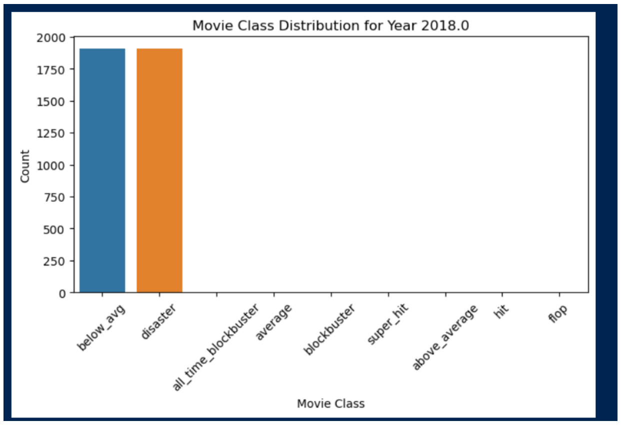
Movie class distribution for each year

What is the movie success distribution for each year?

The distribution of movie success from 1985 to 2012 reveals fluctuations, with a prevalence of movies categorized as "Below Average" and "Disaster." Noteworthy periods include a majority of underperforming movies between 1994 and 1997 and a well-distributed performance across all classifications from 2007 to 2010. From 2012 to 2016, a mix of "Below Average," "Disaster," and "All-Time Blockbuster" movies is observed, while 2018 to 2023 displays a balanced distribution between "Below Average" and "Disaster." The majority of movies fall into the categories of "Below Average" and "Disaster," with a limited number of "All-Time Blockbusters" and minimal occurrences of flops, hits, and averages.



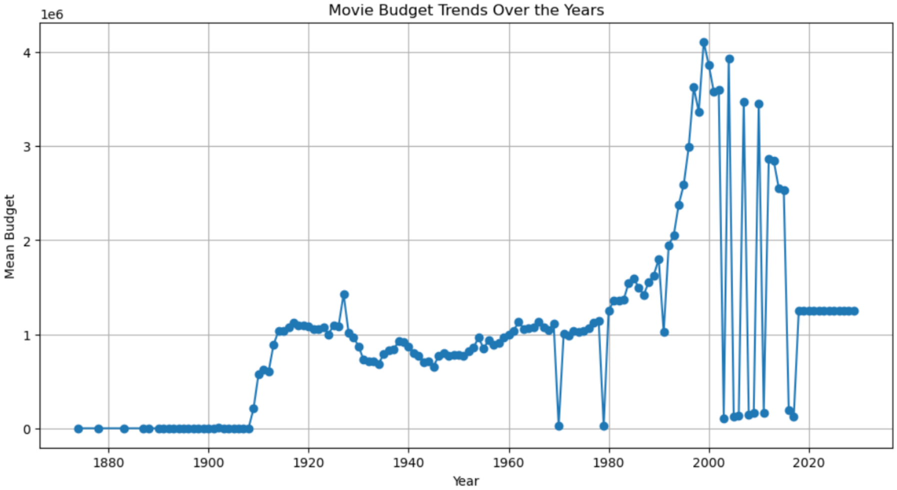




Yearly investment trend

What is the yearly investment trend in movies?

Investment in the film industry exhibited substantial growth over the years, reaching its peak around 2000 before a decline in 2020, potentially attributed to the impact of the coronavirus outbreak. Notably, there was a slight peak in investment around 1925, coinciding with pivotal events such as the transition from silent films to "talkies" with synchronized sound. This period also witnessed the revolutionary development in cinema and the significant ascendancy of Hollywood.

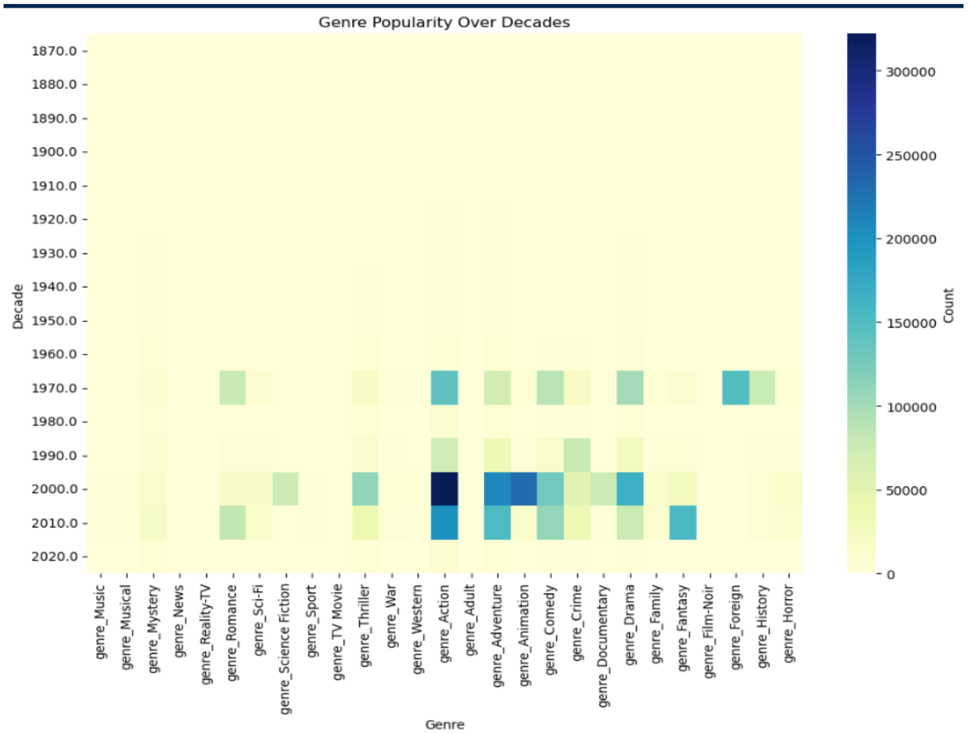


Popularity of genres over the years

Which genre is popular each year?

From 1970 onwards, there is an observable upward trend in the production of Action movies, peaking in 2000 with the release of numerous Marvel and DC films. Simultaneously, Adventure and Animation genres show an increasing popularity, while Foreign and History genres experience a decline.

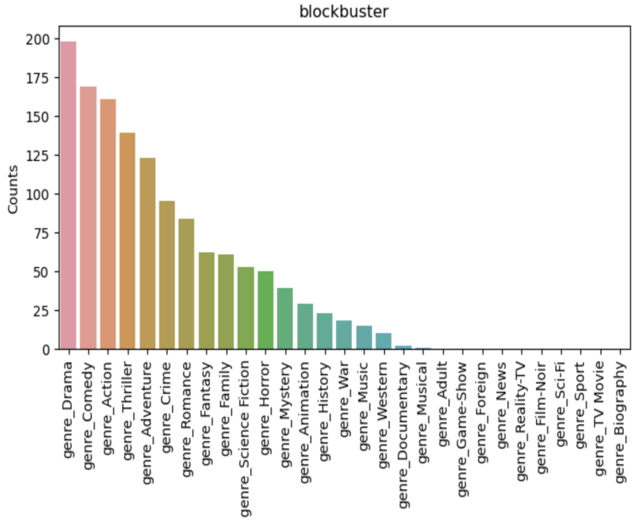
The dominant genres identified are Action, Adventure, Animation, Fantasy, and Comedy. In essence, this project aims to conduct a thorough analysis of the film industry, offering insights into the determinants of a film's success. Through the utilization of two extensive datasets and established methodologies, the objective is to uncover valuable knowledge about the intricacies of the film industry.

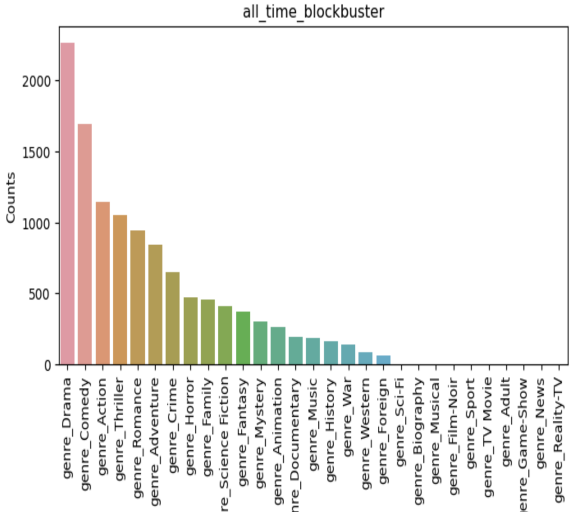


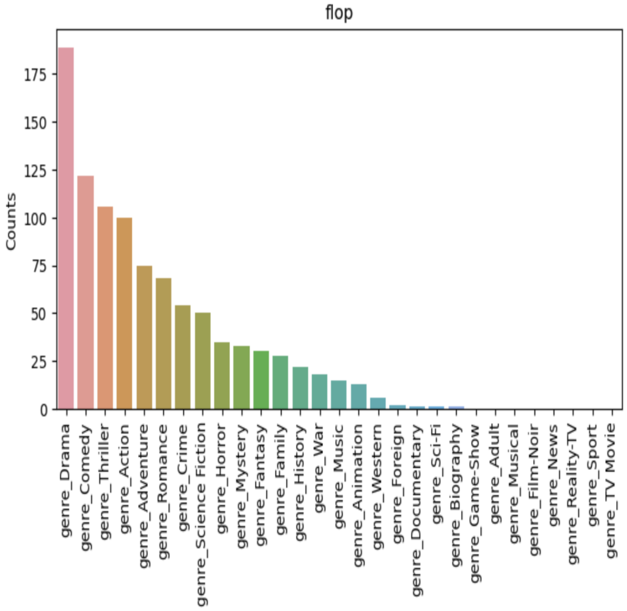
Role of Genre in Movie Success

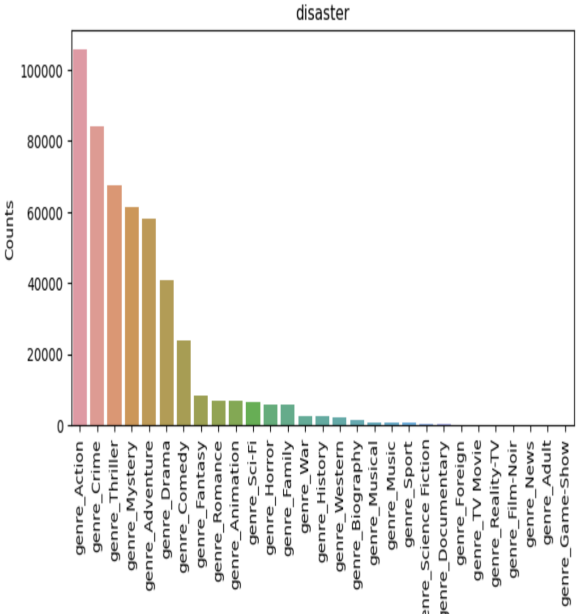
How does the movie genre contribute to movie success?

Genre distribution has been scrutinized to evaluate its impact on the success of films, particularly focusing on the four extreme categories: "All-Time Blockbuster," "Blockbuster," "Flop," and "Disaster."

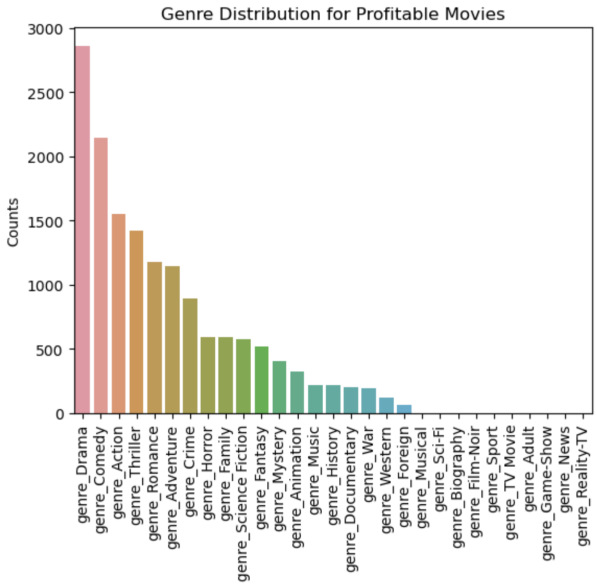


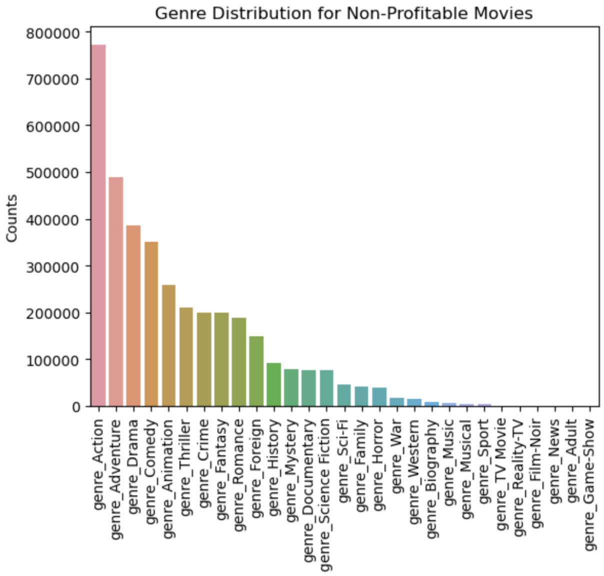






Analysis has been conducted on the distribution of genres for two categories: movies that were generated and movies that were not generated. This examination delves into the diversity of genres within these distinct sets of films.

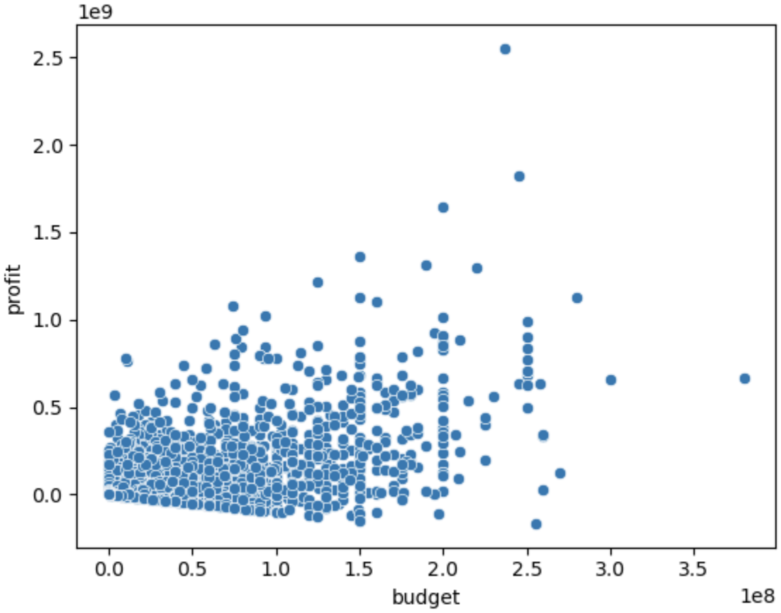




Budget vs Profit

How does the movie budget contribute to movie success?

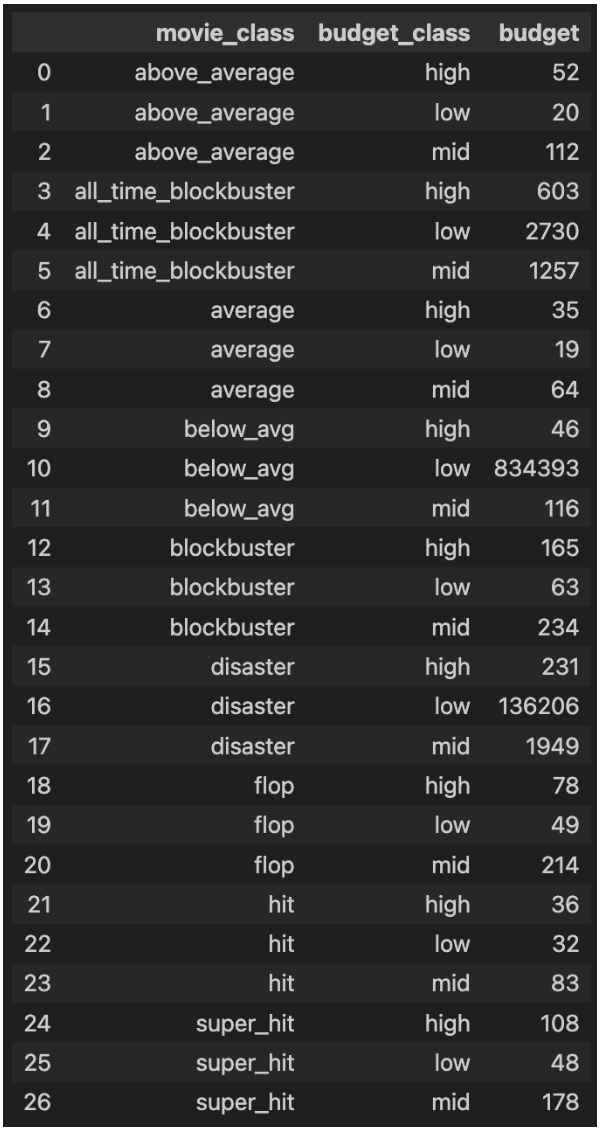
The examination of the relationship between allocated movie budgets and generated profits indicates that while budget is not the sole determinant of a film's success, there is a correlation between budget size and profit. While acknowledging that budget is not the primary factor influencing success, it is evident that for certain films, an increase in budget corresponds to higher profits, suggesting a partial contribution of budget size to overall profitability.



Budget vs Movie Success

How does the movie budget contribute to movie success?

Our efforts included calculating the budget for each film category. The graphical representation reveals that a substantial number of films categorized as "below average" and "disasters" tend to have relatively low budgets. This observation aligns with the common understanding that producing a high-quality film often necessitates a substantial financial investment.



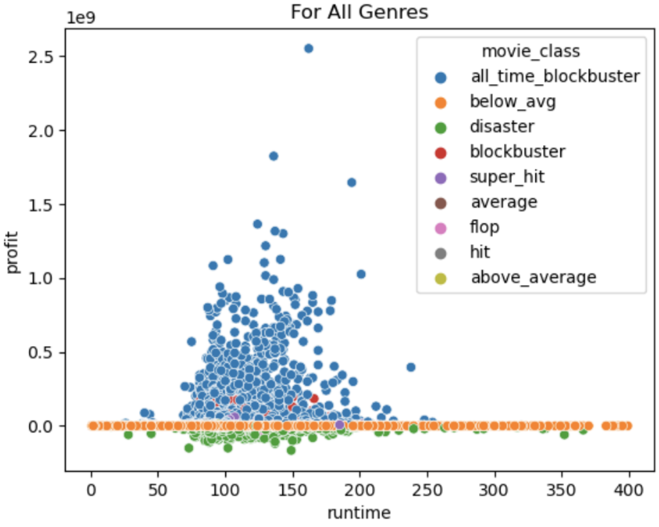
	movie_class	budget_class	budget
0	above_average	high	52
1	above_average	low	20
2	above_average	mid	112
3	all_time_blockbuster	high	603
4	all_time_blockbuster	low	2730
5	all_time_blockbuster	mid	1257
6	average	high	35
7	average	low	19
8	average	mid	64
9	below_avg	high	46
10	below_avg	low	834393
11	below_avg	mid	116
12	blockbuster	high	165
13	blockbuster	low	63
14	blockbuster	mid	234
15	disaster	high	231
16	disaster	low	136206
17	disaster	mid	1949
18	flop	high	78
19	flop	low	49
20	flop	mid	214
21	hit	high	36
22	hit	low	32
23	hit	mid	83
24	super_hit	high	108
25	super_hit	low	48
26	super_hit	mid	178

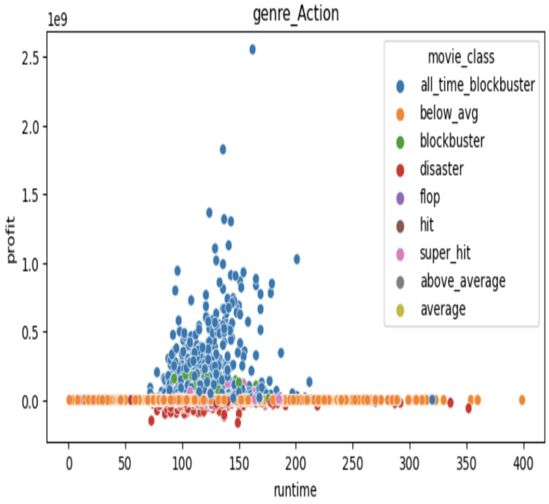
Movie Runtime vs Profit

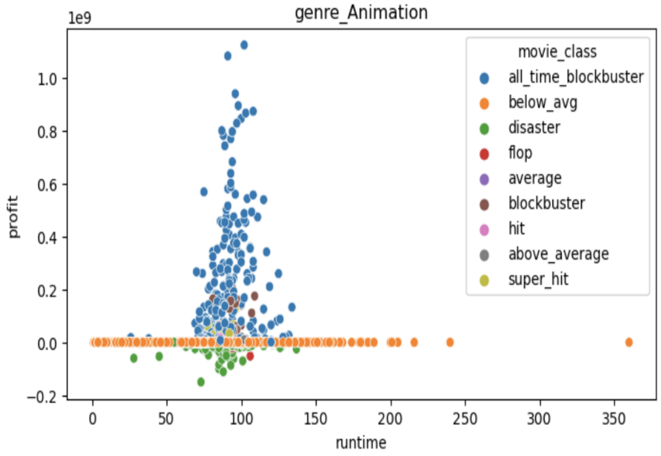
Does the movie runtime contribute to profit?

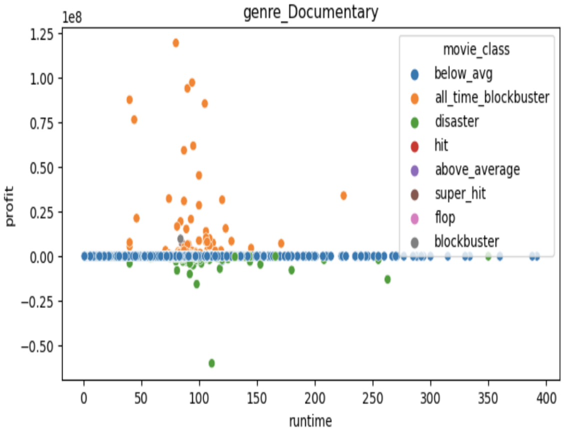
Subsequently, scatter plots have been generated to depict the relationship between movie runtimes and profits. Separate scatter plots have been created for each genre, providing a visual representation of the correlation between the duration of movies and the profits they generated within specific genres.

In general, it is inferred that a movie's runtime should fall within a certain range for optimal success. Additionally, for certain genres, there is a notable correlation between movie runtimes and profits, suggesting that the duration of films within these genres is highly influential on their financial success.



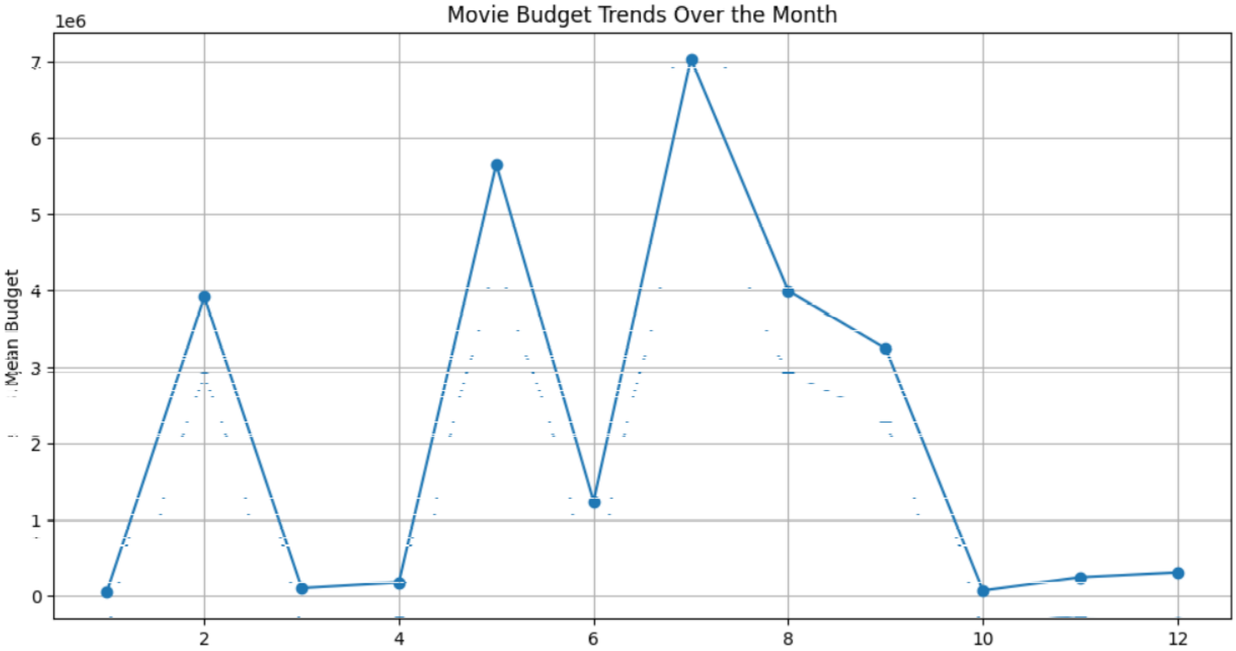






Analysis of Movie Budget through the months

How is the movie budget through the months

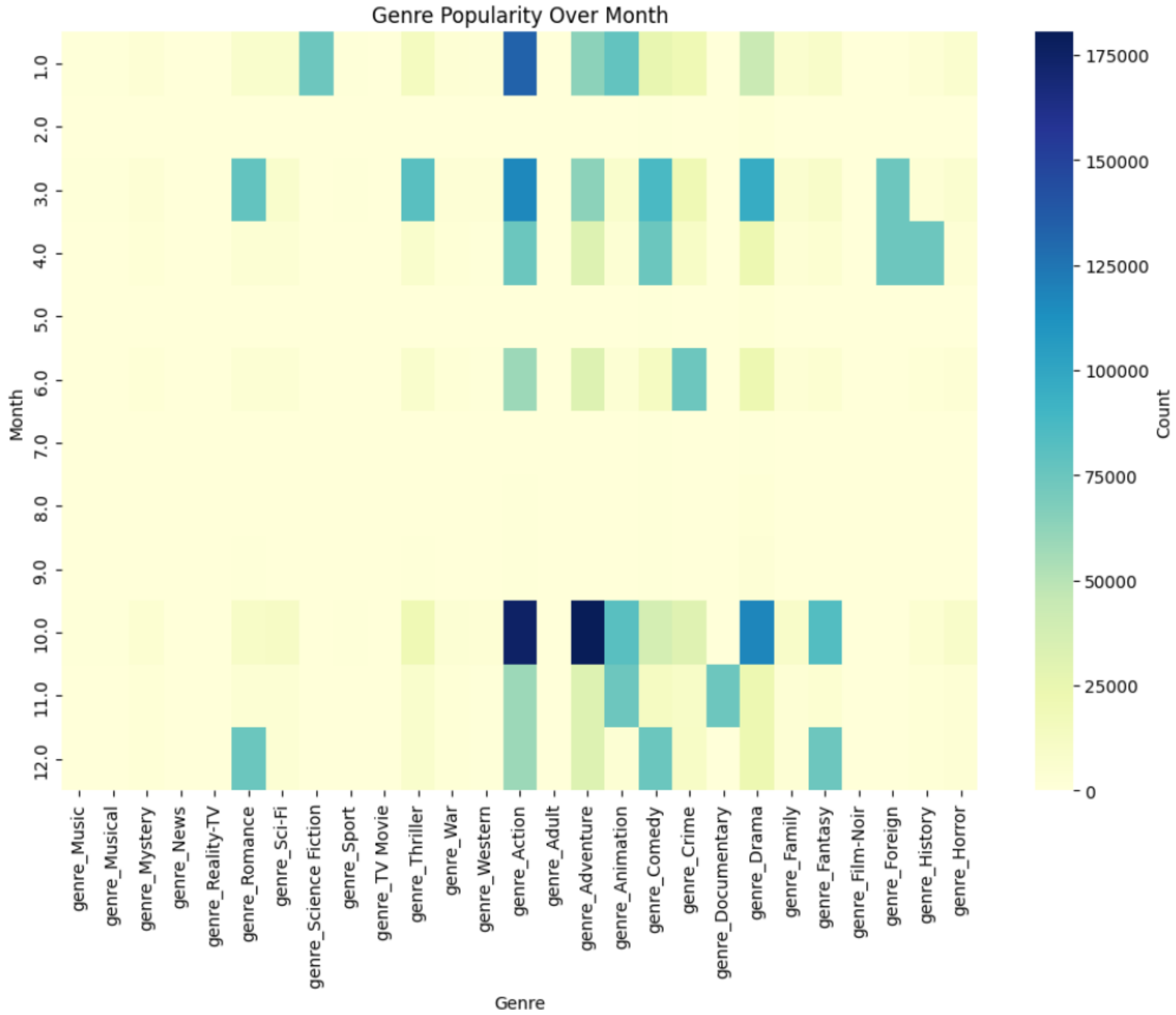


The plotted data reveals trends in movie budgets across the months. Movies released in July exhibit the highest budgets, while those released in January appear to have the lowest budgets, potentially influenced by reduced audience attendance during vacations. This pattern aligns with the assumption that, in January, fewer people visit cinemas due to vacation-related factors, while July, with increased leisure time, becomes a target period for big-budget releases.

Popularity of genres across the months

Which genre is popular for each month?

A specific movie genre appears to be popular for each month, with Action and Adventure genres being particularly favored, especially in October. Additionally, Dramas also gain popularity during October. Action movies maintain popularity in January and March, while Romance movies dominate in January. Consequently, targeting specific genres for release months is recommended, such as releasing Action and Adventure movies in October. This analysis establishes a foundation for comprehending the correlation between movie release months, genre choices, and their potential success probabilities.



Feature Engineering

Computation of Star Power

The influence of cast and crew on a movie's success is evident, with actors like Tom Cruise and Shahrukh Khan being notable examples in Hollywood and Bollywood, respectively. Consequently, we decided to quantify the star power and director power based on the provided cast and crew information. The approach involved collecting the total movie revenues associated with each star and director, summing them to derive star_power and director_power columns. These computed values serve as useful features for our model-building process.

Classification of movie gross collection based on the reference

https://balusboxoffice.quora.com/Classification-of-Movies-as-Hits-Superhits-Blockbusters-Flops

Movies were categorized based on their profit percentages according to the criteria outlined in the above reference.

- Profit 0% - 10%: Average
- Profit >10% up to 25%: Above Average
- Profit > 25% up to 40%: HIT
- Profit > 40% up to 75%: SUPERHIT
- Profit > 75% up to 125%: BLOCKBUSTER
- Profit > 125%: ALL TIME BLOCKBUSTER
- Loss of Less than 15%: BELOW AVERAGE
- Loss of More than 15% but less than 40%: FLOP
- Loss of More than 40%: DISASTER

This classification provides a framework for understanding the financial success or failure of movies based on their profit margins.

Extraction of year and month from Release Year

The process involved extracting the release year and month from the release date, creating new columns for this information. This extraction facilitated a more straightforward visualization of the data, enabling the identification of patterns and enhancing understanding of the temporal effects on movie profits.

Data Pre-processing and Feature Selection

Feature Selection

Following our Exploratory Data Analysis (EDA), key predictors such as 'budget,' 'release_year,' 'release_month,' 'runtime,' 'certificate,' 'star_power,' and 'director_power' were identified. These features were designated as X, while the target label 'movie_class' was set as Y. The dataset was then partitioned into training and testing sets (X_train, X_test, y_train, y_test) with an 80:20 ratio for subsequent analysis and model training.

Data Preprocessing

For data preprocessing, a pipeline was employed to efficiently handle null values and potential data leakage, enabling parallel execution of tasks. Kernel Density Estimation (KDE) plots were utilized to guide the imputation and scaling strategies for various columns.

Numeric transformers such as 'budget,' 'director_power,' and 'star_power' underwent imputation using an Iterative Imputer due to their dependency on other movie-related factors. This decision was made to capture the interactive and dependent relationships between these features, avoiding the use of simple imputers with mean or median values.

Observing the skewed distribution of 'budget,' 'director_power,' and 'star_power,' Min-Max scaling was applied to standardize these values within a specific range. This addressed the impact of skewed distributions.

For 'release_month' and 'release_year,' which exhibited distributions around the mean, Standard Scaler was applied to standardize the features by removing the mean and scaling to unit variance.

Handling null values in categorical columns, specifically 'certificate,' involved filling missing values with "Not Rated" and creating a new unique column value. Subsequently, one-hot encoding was applied to handle the lack of ordinality and the relatively small number of unique values.

In summary, the pipeline integrates an Iterative Imputer for numeric transformers, Min-Max scaling for skewed distributions, Standard Scaler for distributions around the mean, and appropriate strategies for handling null values and categorical features.

Modeling

Model Selection

To choose the optimal model, we employed Cross-Validation to ensure suitability for our use case. Addressing class imbalances in the dataset, StratifiedKFold was applied. Key observations include:

Algorithm Name	Accuracy	F1 Macro Score	F1 Weighted Score
LogisticRegression	0.88	0.14	0.84
RandomForestClassifier	0.993 (best)	0.297 (best)	0.992 (best)
KNeighbour Classifier	0.989	0.268	0.988
Gradient Boosting Classifier	0.991	0.245	0.990

Discussions:

Logistics Regression:

Logistic Regression, being a linear model primarily used for binary classification, may face limitations in capturing complex relationships or handling class imbalances. The lower F1 Macro Score suggests potential struggles with minority classes or imbalanced data. The high accuracy might be influenced by a bias towards the majority class.

Random Forest Classifier:

Random Forest, as an ensemble method, excels in capturing complex relationships and handling outliers. The high accuracy and F1 Weighted Score indicate strong overall performance. However, the lower F1 Macro Score suggests possible challenges with certain minority classes, emphasizing the need for further investigation into imbalanced class performance.

K Neighbors Classifier:

The K Neighbors Classifier, relying on proximity-based decision-making, demonstrates high accuracy and F1 Weighted Score, indicating good overall performance. However, the lower F1 Macro Score suggests potential difficulties with minority classes, hinting at challenges in classifying less prevalent categories.

Gradient Boosting Classifier:

Gradient Boosting, another ensemble method, builds trees sequentially to emphasize misclassified instances. The high accuracy and F1 Weighted Score point to strong overall performance. The lower F1 Macro Score, however, indicates potential challenges in handling imbalanced classes or capturing nuanced relationships.

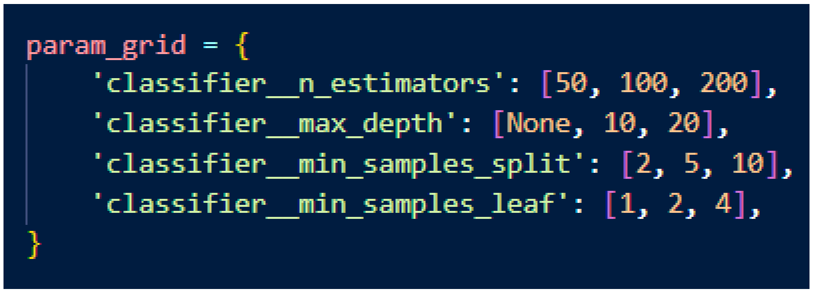
Possible Explanations for F1 Macro Score:

Imbalanced Data: F1 Macro Score is sensitive to minority class performance. Imbalanced datasets may lead to difficulty generalizing to minority classes, resulting in lower F1 Macro Scores.

Feature Limitations: The features used for classification might not effectively capture distinctions between different classes, impacting the macro-level F1 scores.

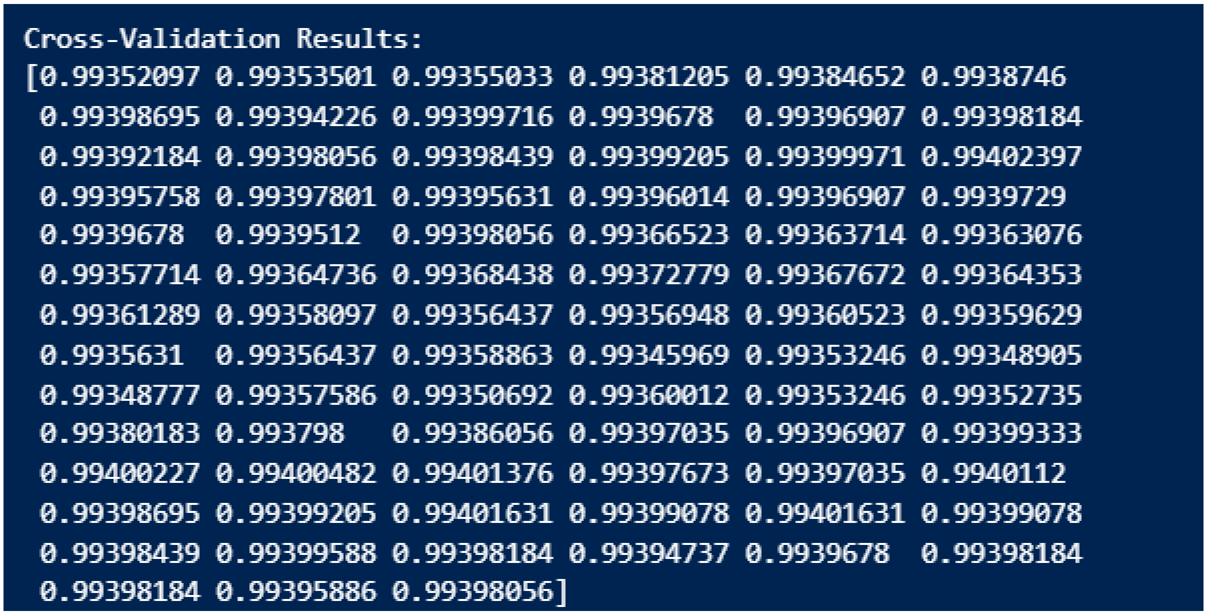
Hyperparameter Tuning

We then proceeded to Grid Search the best parameters for the Random Forest. We had the following parameters:



```
param_grid = {
    'classifier__n_estimators': [50, 100, 200],
    'classifier__max_depth': [None, 10, 20],
    'classifier__min_samples_split': [2, 5, 10],
    'classifier__min_samples_leaf': [1, 2, 4],
}
```

And observed the following results:



```
Cross-Validation Results:
[0.99352097 0.99353501 0.99355033 0.99381205 0.99384652 0.9938746
 0.99398695 0.99394226 0.99399716 0.9939678  0.99396907 0.99398184
 0.99392184 0.99398056 0.99398439 0.99399205 0.99399971 0.99402397
 0.99395758 0.99397801 0.99395631 0.99396014 0.99396907 0.9939729
 0.9939678  0.9939512  0.99398056 0.99366523 0.99363714 0.99363076
 0.99357714 0.99364736 0.99368438 0.99372779 0.99367672 0.99364353
 0.99361289 0.99358097 0.99356437 0.99356948 0.99360523 0.99359629
 0.9935631  0.99356437 0.99358863 0.99345969 0.99353246 0.99348905
 0.99348777 0.99357586 0.99350692 0.99360012 0.99353246 0.99352735
 0.99380183 0.993798   0.99386056 0.99397035 0.99396907 0.99399333
 0.99400227 0.99400482 0.99401376 0.99397673 0.99397035 0.9940112
 0.99398695 0.99399205 0.99401631 0.99399078 0.99401631 0.99399078
 0.99398439 0.99399588 0.99398184 0.99394737 0.9939678  0.99398184
 0.99398184 0.99395886 0.99398056]
```

After conducting a Grid Search for the best parameters for the Random Forest model, the following hyperparameters were determined:

- max_depth: None (indicating nodes are expanded until they contain fewer than min_samples_split samples)
- min_samples_leaf: 2 (the minimum number of samples required to be at a leaf node)

- min_samples_split: 10 (the minimum number of samples required to split an internal node)
- n_estimators: 200 (the number of trees in the forest)

The best cross-validation score obtained was 0.994, indicating a high level of performance on the validation sets.

Interpretation:

- Robust Performance: The consistently high cross-validation scores across different folds suggest that the Random Forest model is robust and performs well on various subsets of the training data.
- Effective Hyperparameters: The chosen hyperparameters, as indicated by the "Best Parameter Combination," appear to be effective for the given task. The model achieves an exceptionally high overall cross-validated score of 0.994, reflecting its capability to generalize well to unseen data.

This outcome reinforces the suitability of the Random Forest model for the classification task at hand, with the selected hyperparameters demonstrating strong performance across different subsets of the training data.

ANN

In our attempt to explore different approaches, we ventured into defining, compiling, and training a neural network model for a classification task using the Keras library. This task specifically involves predicting one of nine classes. We designed a sequential neural network model with multiple dense layers, culminating in a softmax output layer. The model was compiled using the Adam optimizer, categorical cross-entropy loss, and additional metrics, including accuracy and TensorFlow's recall metric. The training process spanned 100 epochs with a batch size of 204800.

Results and Discussion

We employed classification reports to assess the performance of two distinct models—Artificial Neural Network (ANN) and Random Forest—on a multiclass classification task.

Artificial Neural Network (ANN)

Overall Performance

- Accuracy: 96%
- Weighted Average F1-Score: 98%
- Macro Average F1-Score: 27%

Class-Specific Metrics

Class 0 ("all_time_blockbuster"):

Low precision (2%) and recall (16%).

Low F1-score (3%).

Class 1 ("blockbuster"):

Precision (8%) and recall (54%) are relatively low.

F1-score (15%) is also modest.

Class 2 ("super_hit"):

Precision, recall, and F1-score are all very low (0%).

Class 3 ("hit"):

High precision (100%) and recall (97%).

High F1-score (98%)

Class 4 ("above_average"):

Precision (7%) and recall (24%) are both low.

F1-score (11%) is relatively low.

Class 5 ("average"):

High precision (100%) and recall (97%).

High F1-score (98%).

Class 6 ("below_avg"):

Precision (5%) and recall (26%) are relatively low.

F1-score (9%) is modest.

Class 7 ("flop"):

Low precision (1%) and recall (3%).

Very low F1-score (1%).

Class 8 ("disaster"):

Precision (4%) and recall (19%) are both low.

F1-score (6%) is relatively low.

Summary of Artificial Neural Network (ANN) Model Performance

The ANN model exhibits high overall accuracy and weighted F1-score, largely influenced by outstanding performance in the majority class (Class 3). However, challenges arise in handling minority classes (Classes 0, 2, 4, 6, 7, 8), indicated by low precision, recall, and F1 scores for these specific classes.

Random Forest (with Best Parameters)

Overall Performance

Accuracy: 99%

Weighted Average F1-Score: 99%

Macro Average F1-Score: 26%

Class-Specific Metrics

Class 0 ("all_time_blockbuster"):

Precision, recall, and F1-score are all very low (0%).

Class 1 ("blockbuster"):

Precision (43%) and recall (24%) are moderate.

F1-score (30%) is relatively higher compared to ANN.

Class 2 ("super_hit"):

Precision, recall, and F1-score are all very low (0%).

Class 3 ("hit"):

High precision (100%) and recall (100%).

High F1-score (100%).

Class 4 ("above_average"):

Precision, recall, and F1-score are all very low (0%).

Class 5 ("average"):

High precision (98%) and recall (99%).

High F1-score (99%).

Class 6 ("below_avg"):

Precision (50%) is relatively higher.

Recall (1%) and F1-score (3%) are low.

Class 7 ("flop"):

Precision, recall, and F1-score are all very low (0%).

Class 8 ("disaster"):

Precision, recall, and F1-score are all very low (0%).

Summary of Random Forest Model Performance

The Random Forest model surpasses the ANN across various metrics, achieving higher precision, recall, and F1-scores for several classes. This performance contributes to excellent overall accuracy and weighted F1-score. Similar to the ANN, the Random Forest model faces challenges with minority classes (Classes 0, 2, 4, 6, 7, 8).

Comparison of Model

Comparison of Artificial Neural Network (ANN) and Random Forest Models:

Model Complexity

ANN: Neural networks are highly flexible and capable of capturing complex relationships in data. However, this flexibility might lead to overfitting, especially in cases with imbalanced classes.

Random Forest: Random Forest, being an ensemble of decision trees, tends to handle complex relationships well and is less prone to overfitting. The aggregation of multiple trees enhances robustness.

Sensitivity to Initialization and Hyperparameters

ANNs are sensitive to the choice of initial weights and hyperparameters.

Improper tuning may result in challenges in generalizing to minority classes, leading to lower precision, recall, and F1 scores.

Data Imbalance

Both Models: The struggle with minority classes in both models (Classes 0, 2, 4, 6, 7, 8) suggests that imbalanced data might pose challenges. Insufficient representation of minority classes can impact model generalization.

Feature Importance

ANN: Neural networks automatically learn feature representations, and their interpretability can be challenging. Certain features may not be adequately emphasized.

Random Forest: Random Forest provides feature importance scores, making it easier to understand the contribution of each feature to the model's decision. This can be advantageous in understanding which features are critical for classification.

Hyperparameter Tuning

ANN: Hyperparameter tuning for neural networks can be complex, and finding the optimal architecture may require extensive experimentation.

Random Forest: Random Forest is less sensitive to hyperparameters, and default settings often provide reasonable performance. This can simplify the model selection process.

Ensemble vs. Single Model

ANN: A single neural network is employed, potentially limiting its ability to capture diverse patterns in the data.

Random Forest: Being an ensemble of decision trees, the Random Forest can collectively consider different aspects of the data, contributing to improved generalization.

Training Speed

ANN: Training deep neural networks can be computationally intensive and time-consuming.

Random Forest: Random Forest training is generally faster, making it more feasible for certain applications.

In summary, the choice between the ANN and Random Forest models depends on the specific characteristics of the dataset, the interpretability of results, and the trade-off between model complexity and computational efficiency. Both models exhibit strengths and limitations, and the choice should align with the goals and requirements of the classification task.

Recommendations:

Hyperparameter Tuning for ANN

Fine-tune the architecture and hyperparameters of the ANN to optimize its performance, especially considering the class imbalance.

Data Augmentation or Resampling

Explore techniques such as data augmentation or resampling to balance class distribution, aiding both models in learning patterns from minority classes.

Feature Importance Analysis

Conduct a feature importance analysis to identify influential features. This can guide feature engineering efforts and improve the models' understanding of the data.

Ensemble Strategies

Consider combining predictions from both models using ensemble strategies. This can leverage the strengths of each model and potentially enhance overall performance.

Threshold Adjustment

Experiment with adjusting classification thresholds to balance precision and recall based on the specific goals of the classification task.

Cross-Validation and Robust Evaluation

Ensure that model evaluation is robust through techniques like cross-validation to provide a more comprehensive assessment of performance.

By addressing these considerations and iteratively refining the models, it's possible to enhance their capabilities and achieve better performance, particularly in minority classes.

Future Work

Recommendation System for Genre-Specific Features

Develop a recommendation system that tailors suggestions for genre-specific star power, runtime, and release month. This could involve collaborative filtering, content-based filtering, or hybrid methods to provide personalized recommendations based on user preferences and historical data.

NLP for Movie Plot Analysis

Implement Natural Language Processing (NLP) techniques to analyze movie plots. This could involve sentiment analysis, topic modeling, or even building a text-based model to extract valuable information from plot summaries. Insights from NLP could contribute to a better understanding of audience preferences and predict movie success.

Data Collection for Rare Classes

Collect additional data, specifically targeting rare classes in your classification task. This could involve actively seeking out and including examples of movies that fall into rare classes. Increased representation of rare classes in the dataset will contribute to more robust model training and better generalization.

Temporal Analysis for Release Month

Explore temporal analysis specifically related to release months. This could involve identifying patterns or trends in movie success based on the time of year. Understanding seasonal effects on audience preferences can contribute to more informed decision-making in the film industry.

Collaboration with Industry Experts

Collaborate with industry experts, such as film critics, producers, or marketing professionals, to gather insights into the factors influencing movie success. This collaboration can provide valuable domain knowledge that can guide feature engineering and model improvement.

User Feedback and Iterative Improvement

Implement mechanisms for collecting user feedback on movie recommendations or predictions. This feedback loop can be used for iterative model improvement, ensuring that the system continuously adapts to changing user preferences and industry dynamics.

Explainability for Recommendations

Incorporate explainability into the recommendation system. Users are more likely to trust and engage with a system that provides transparent explanations for its recommendations. Techniques such as SHAP values or attention mechanisms can be useful for this purpose.

Continuous Model Monitoring

Establish a system for continuous monitoring of model performance, especially as new data becomes available. Periodically retrain the model to ensure it remains relevant and effective in capturing evolving patterns in the movie industry.

Ensemble Modeling for Recommendations

Consider building ensemble models for recommendation by combining predictions from multiple models. This can enhance the diversity and accuracy of recommendations, especially when dealing with different types of features (numerical, categorical, text).

Evaluation Metrics for Recommendation System

Define and use appropriate evaluation metrics for the recommendation system. Metrics such as precision, recall, and Mean Average Precision (MAP) are common for recommendation systems. Customize the metrics based on the specific goals and characteristics of your application.

User Personalization

Extend the recommendation system to incorporate user personalization. This could involve building user profiles based on historical preferences and adapting recommendations accordingly.

References

Dataset link:

IMDb Movies Dataset

https://www.kaggle.com/datasets/rajugc/imdb-movies-dataset-based- on-genre)

TMDB Dataset

https: //www.kaggle.com/datasets/rounakbanik/the-movies-dataset