Is the Movie a Hit or a Flop?

Project Report

Team 6B, Dream Team:

Naomi Hossain

Xinyi He

Luisa Garate

Weinou Jin

Tysean Haigood

Table of Contents

1. Executive summary

The goal of this project was aimed at creating a predictive model through machine learning to determine whether a movie is a hit or a flop. The dataset we used was obtained from Kaggle, named IMDb Movies Dataset. This data set contains 10,178 rows of movies with a mix of numerical, categorical, and text-based variables. We were able to create a profit-based target variable and prepare data through cleaning, encoding, and feature transformation. We created two classification techniques. These were Naïve Bayes and Decision Tree. Our findings can be able to help businesses make better decisions in how they invest in movies. We found that high budget films with a strong audience are able to succeed and hit, while in comparison, low-budget films tend to flop.These insights can be able to help studios allocate resources correctly. We recommend that studios invest and prioritize films that have a stronger budget and strong audience potential. Overall, this analysis shows us how data-driven decisions can help make informed decisions and make profitable decision making in the film industry. This project was created by team 6B also known as 'Dream Team'.

2. Introduction: Explain the business idea, why it's important, and the data source

2.1 Introduction

The entertainment industry is one of the largest and most influential industries in today's economy. Especially in Southern California, we are surrounded by constant exposure to media from movies and television shows to music, podcasts, and digital streaming content. We wanted to specifically focus on the film industry.

The film industry within entertainment is extremely influential. It generates about $100 billion dollars and is expected to grow to $169 billion by 2030. The industry drives economic growth through streaming platforms, theatrical releases, marketing campaigns, merchandise, and international distribution. Its impact reaches far beyond the screen, it also is supporting thousands of jobs in directing, acting, visual effects, editing, data analytics, and more. In regions like Southern California, the film industry is especially prominent, with major studios, production companies, and creative talent concentrated in one of the world's entertainment capitals. As technology evolves and audiences consume content in new ways, the film industry continues to expand, reinforcing its role as a powerful cultural and economic force.

Its continued growth depends not only on artistic innovation but also on the strategic use of data to guide production, budgeting, marketing, and distribution decisions. As the industry becomes increasingly data-driven, analytical tools such as predictive modeling offer valuable insights into the factors that contribute to a film's success. By being able to use machine learning techniques, it can be leveraged for studios, analysts, streaming platforms, and investors. It will help better anticipate audience outcomes, allocate resources more efficiently, and reduce financial risk. These types of analysis can provide a more structured approach to understanding which film attribute most strongly impacts box office performance, demonstrating the potential of data science to support evidence-based decision making within the entertainment industry.

2.2 Our Project

In our project, we apply predictive analytics to explore the key factors that contribute to box office performance. Using a decision tree model, we were able to analyze various movie attributes to uncover patterns that differentiate successful films from unsuccessful ones. By visualizing these relationships and examining how the model splits the data, we gain insight into the features that most strongly influence performance. This analysis not only helps to understand the factors behind movie success but also demonstrates how predictive models can support data-driven decision making within the entertainment industry.

2.3 Business Impact/Implications

Predictive modeling in the film industry has significant implications for strategic planning and resource allocation. A decision tree model that classifies movies as "hit" or "flop" based on attributes such as budget, genre, runtime, cast, and release timing provides studios with a structured framework for evaluating risk before a film enters production. By identifying the most influential features associated with commercial success, studios can make more informed decisions about which projects to approve, how much to invest, and where additional resources may be required. This would help studios determine which films are most likely to achieve commercial success, allowing them to prioritize high-potential projects while avoiding investments in movies with a higher likelihood of underperformance (*Zhang 2024*). By evaluating predicted outcomes before committing significant financial resources, studios can better allocate budgets, negotiate contracts, and manage production timelines in ways that maximize expected return on investment.

Beyond production decisions, the model can support marketing and distribution strategies. By analyzing patterns within historical film data, decision makers can anticipate the types of movies likely to perform well during specific seasons, identify which genres require stronger promotional efforts, and tailor marketing budgets to the expected performance of each release. This allows marketing teams to optimize advertising spend, select appropriate promotional channels, and better position films in an increasingly competitive entertainment landscape. Insights derived from the model may also help studios determine ideal release windows and distribution platforms based on predicted audience engagement.

3. Data: Data summary, description, visualization.

3.1 Dataset Summary

The dataset we used was obtained from Kaggle's IMDB Movie Dataset. This dataset contains 10,178 movies with the 12 features we used. The features were mixed with some being numerical, others categorical, and even text-based attributes. They key variables that we used included are the following:

Variable’s Name	Variable’s Description
names	the movie’s title (English release names)
data_x	Movie’s release date
score	Movie’s rating score
genre	A list of the movie’s genre’s, separated by commas
overview	A short description or plot of the movie
crew	Names of key people involved (directors, writers, main, cast)
orig_title	The original movie title
status	The film’s release status
orig_lang	Movie’s original language

budget_x	Movie’s production in USD
revenue	Total worldwide box-office revenue in USD
country	Country where movie info was collected from

We decided on this dataset since after analyzing, we decided it would be an efficient model as it captures different features such as budget, genre, and cast. This would allow us to be able to explore how different factors contribute to movie success.

3.2 Data Assessment and Cleaning

We inspected our data and were able to discover issues within our data. For our ‘Genre’ variable we found that there were 85 missing values. For our ‘Crew’ variable we found 56 missing variables. Moreover, our dataset had mixed data types. Some categorical columns were text strings and needed to be transformed. Lastly, our Budget and Revenue variables contained zero values. This would create a distorted model that relies on financial comparisons. We therefore decided that in order to make our predictive model reliable we had to fix it.

In order to fix our data, we removed movies that had a budget = 0 and revenue = 0 as this would have caused a distortion in the predictive model process. Next, we created a new target variable. Since we did not have a label for success we created a binary variable, *hit or flop ,* based on profitability:

- “Hit” if revenue > budget
- “Flop” revenue ≤ budget

Moreover, after creating our target we decided as a team to drop the revenue column. We did not want the model to essentially “cheat” by seeing the revenue. Therefore, after *hit or flop* labels were created,we removed the revenue column.

We also decided to encode categorical features. Variables such as director, country, original language, and genre were transformed into numerical using encoding strategies that were compatible with machine learning algorithms.

Finally, we made sure as we finally looked at our dataset to make sure it’s clean and that there were no other remaining null variables.

3.3 Target Variable Distribution

After creating our target variable the following distribution classes were found

- Hit: 8253 movies
- Flop: 1925 movies

This shows that there is a class imbalance and it is an important characteristic of the dataset.Hits are predicted well but it does struggle a bit with flops due to imbalance.

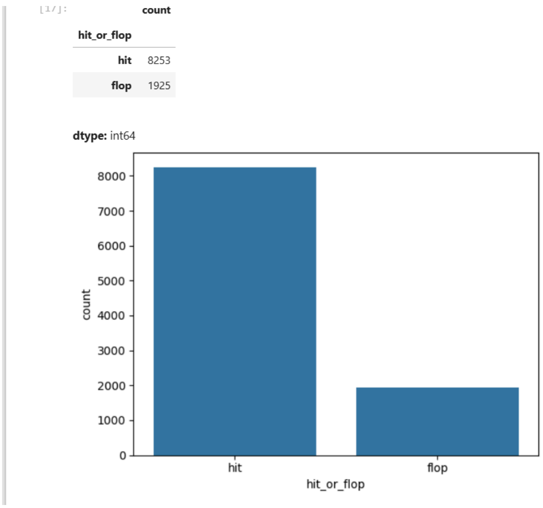
3.4 Feature Engineering

In order to improve predictive performance and provide more meaningful inputs to the model additional steps were taken. Genre was encoded, in order to have multiple genre, they were transformed into binary flags. There was Director and Country encoding, these were transformed into numerical labels. Budget and Score were called for algorithms that are sensitive to magnitude differences. These features allow the model to detect patterns different variables

3.5 Data Visualization

In order to be able to help best understanding our decision-making process, there were several visualizations were created:

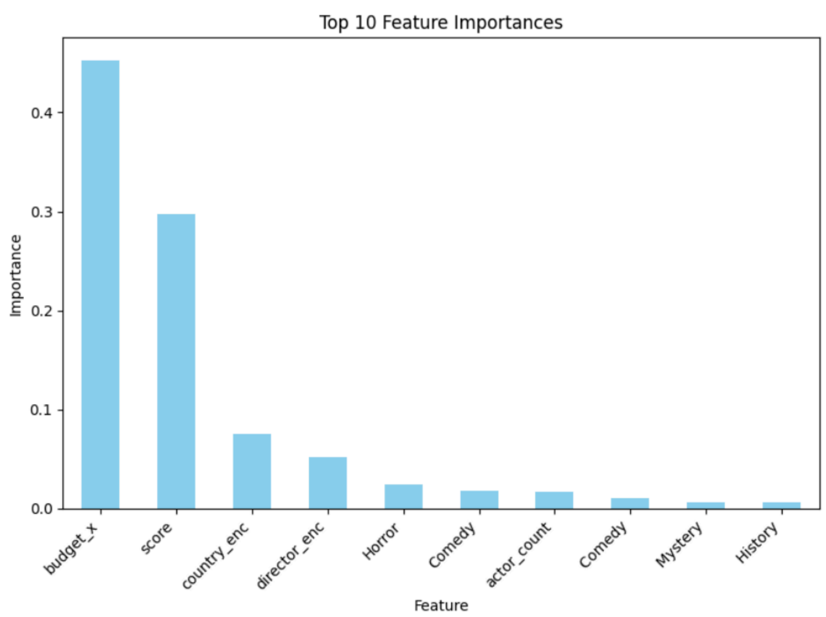
1. Class Distribution Plot



```
[17]:              count
      hit_or_flop
              hit   8253
             flop   1925

      dtype: int64
```

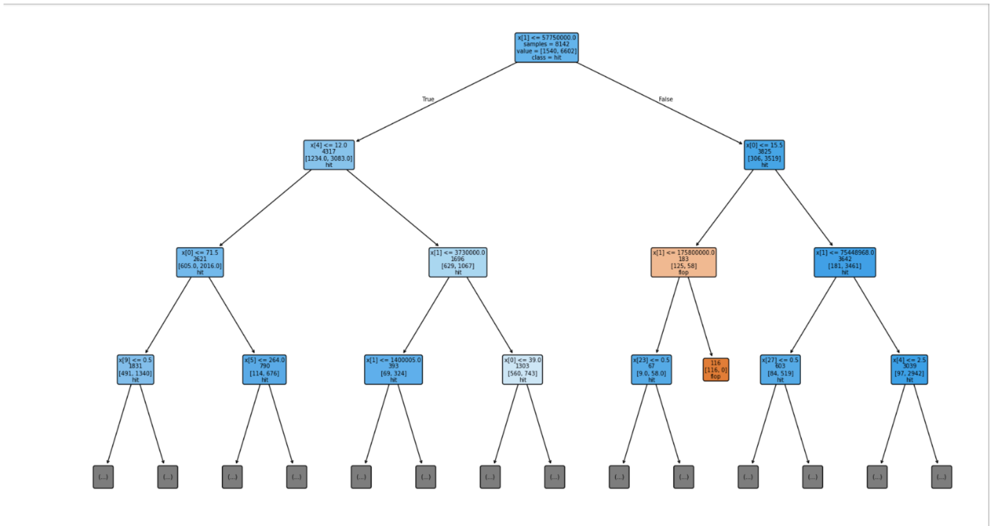
The above visualization shows a bar chart, representing the imbalance between hits and flops.

2. Feature Importance Plot



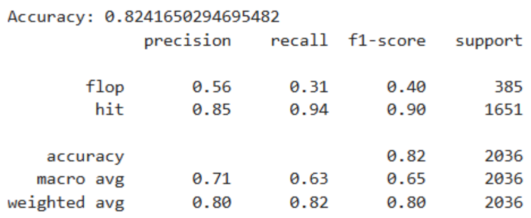
The above visualization shows the ranking of the top predictive features. Budget and score have the most influence followed by the country, director.

3. Decision Tree Visualization

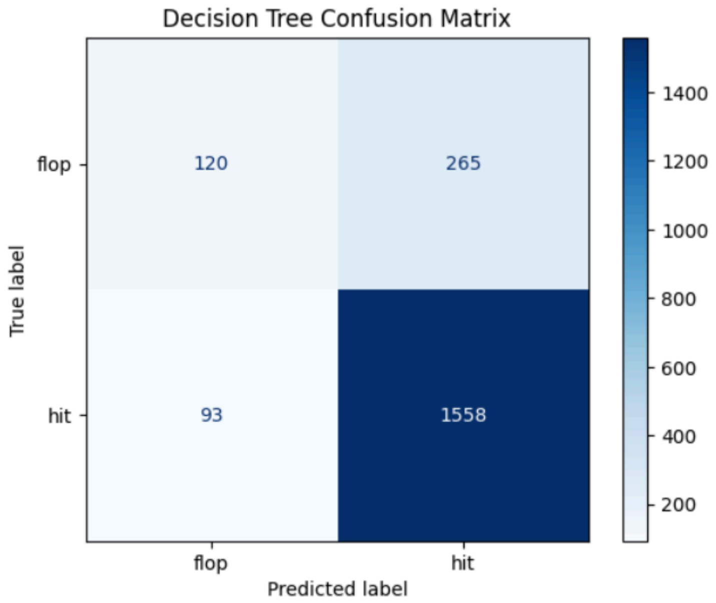


This decision tree visual shows the three levels of the decision tree. The first split was based on budget, demonstrating its strong predictive power.

4. Confusion Matrix



```
Accuracy: 0.8241650294695482
              precision    recall  f1-score   support

        flop       0.56      0.31      0.40       385
         hit       0.85      0.94      0.90      1651

    accuracy                           0.82      2036
   macro avg       0.71      0.63      0.65      2036
weighted avg       0.80      0.82      0.80      2036
```

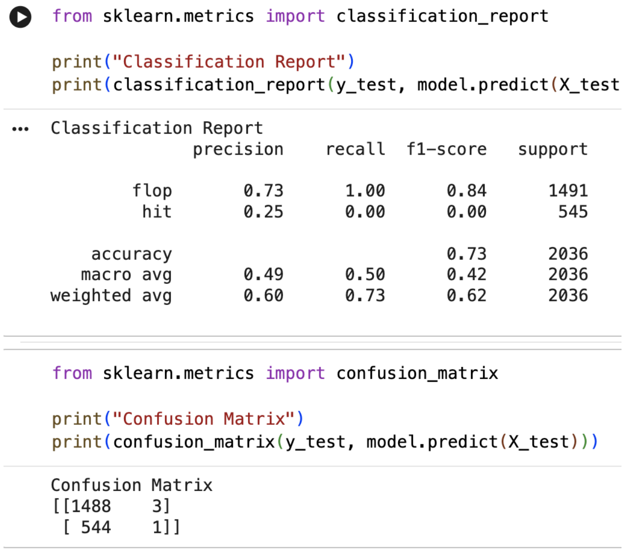


The confusion matrix above predicted how the model hits vs flops. Hits predicted well and flops had more imbalance. This informed modeling strategy and interpretation of results.

4. Analysis

4.1 Naïve Bayes

We began our analysis by experimenting with a Naïve Bayes classifier to predict whether a movie would be categorized as a “hit” or a “flop”. Naïve Bayes is a relatively simple probabilistic model that assumes conditional independence among features. Although this algorithm is computationally efficient and often performs well on high-dimensional datasets, its underlying assumptions presented key limitations for our movie dataset.



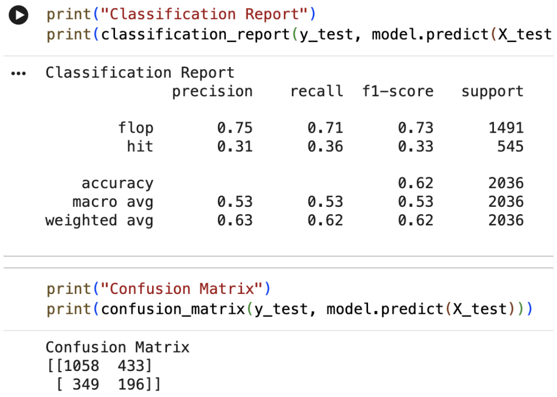
```
from sklearn.metrics import classification_report

print("Classification Report")
print(classification_report(y_test, model.predict(X_test
```

```
Classification Report
              precision    recall  f1-score   support

        flop       0.73      1.00      0.84      1491
         hit       0.25      0.00      0.00       545

    accuracy                           0.73      2036
   macro avg       0.49      0.50      0.42      2036
weighted avg       0.60      0.73      0.62      2036
```

```
from sklearn.metrics import confusion_matrix

print("Confusion Matrix")
print(confusion_matrix(y_test, model.predict(X_test)))
```

```
Confusion Matrix
[[1488    3]
 [ 544    1]]
```

With a Naïve Bayes model, we were only able to use numerical features. So, we were already limited in what features we could use. We build the model on “score” and “budget_x” only. We used Gaussian Naïve Bayes because it is designed for continuous features like what we have in our dataset. The accuracy resulted from the train dataset was 73% and the test data had 73% accuracy as well. This model resulted in a confusion matrix of 1488 true positives, 3 false positives, 544 false negatives, and 1 true negatives. This translates to 1488 flops correctly predicted as flops, 3 flops incorrectly predicted as hits, 544 hits incorrectly predicted as flops, and 1 hits correctly predicted as hits. The model strongly predicts flops, but struggles greatly at predicting hits because hits are a minority in the dataset based on features we have chosen. The classification report resulted in a very low recall for hits because the dataset is imbalanced.



```
print("Classification Report")
print(classification_report(y_test, model.predict(X_test
```

```
Classification Report
              precision    recall  f1-score   support

        flop       0.75      0.71      0.73      1491
         hit       0.31      0.36      0.33       545

    accuracy                           0.62      2036
   macro avg       0.53      0.53      0.53      2036
weighted avg       0.63      0.62      0.62      2036
```

```
print("Confusion Matrix")
print(confusion_matrix(y_test, model.predict(X_test)))
```

```
Confusion Matrix
[[1058  433]
 [ 349  196]]
```

To improve this model we decided to use SMOTE so the classifier learns hits better. After using SMOTE, the model had a confusion matrix of 1058 flops correctly predicted as flops, 433 flops incorrectly predicted as hits, 349 hits incorrectly predicted as flops, and 196 hits correctly predicted as hits. This model had an accuracy score of 62%. Comparing the models, the Naive Bayes model using SMOTE predicts hits much better (from 1 to 196 correct) but it missclassifies more flops as hits (from 3 to 433 false positives). In addition, the overall accuracy of the model also decreased. The model did learn hits better, but the trade-off is more mistakes on flops.

4.2 Naïve Bayes Limitations

Our first limitation was that many of the predictors in our dataset were continuous variables of different types with a nonlinear relationship. These variables included were: budget, runtime, revenue-based ratios, and audience scores. Naïve Bayes does not naturally capture interactions between such features, and its assumption that each variable independently contributes to the outcome does not reflect how movie success operates in reality (*GeeksforGeeks, "Naive Bayes Classifiers"*). For example, budget and genre often interact, and release timing can influence the effectiveness of marketing spend or cast recognition. These dependencies violate the independence assumption and negatively impacted model performance.

Secondly, the dataset included several categorical variables with multiple levels (e.g., genre, production company, cast features). While Naïve Bayes can incorporate categorical predictors, the large number of unique values increased sparsity and reduced the reliability of probability estimates. As a result, the Naïve Bayes model produced lower accuracy and less interpretable outputs, indicating that it was not well-suited for capturing the complex patterns required to classify hit versus flop movies.

Given these limitations, we determined that Naïve Bayes was not the optimal model for this dataset and shifted our approach toward a more flexible, non-parametric algorithm.

4. 3 Decision Tree

To better capture interactions among features, we implemented a decision tree classifier using the movies dataset. Decision trees have a greater advantage for our dataset because they do not require assumptions of linearity or independence. Instead, the model recursively splits the data based on the features that provide the highest information gain, allowing it to uncover nonlinear relationships and meaningful interactions between predictors.

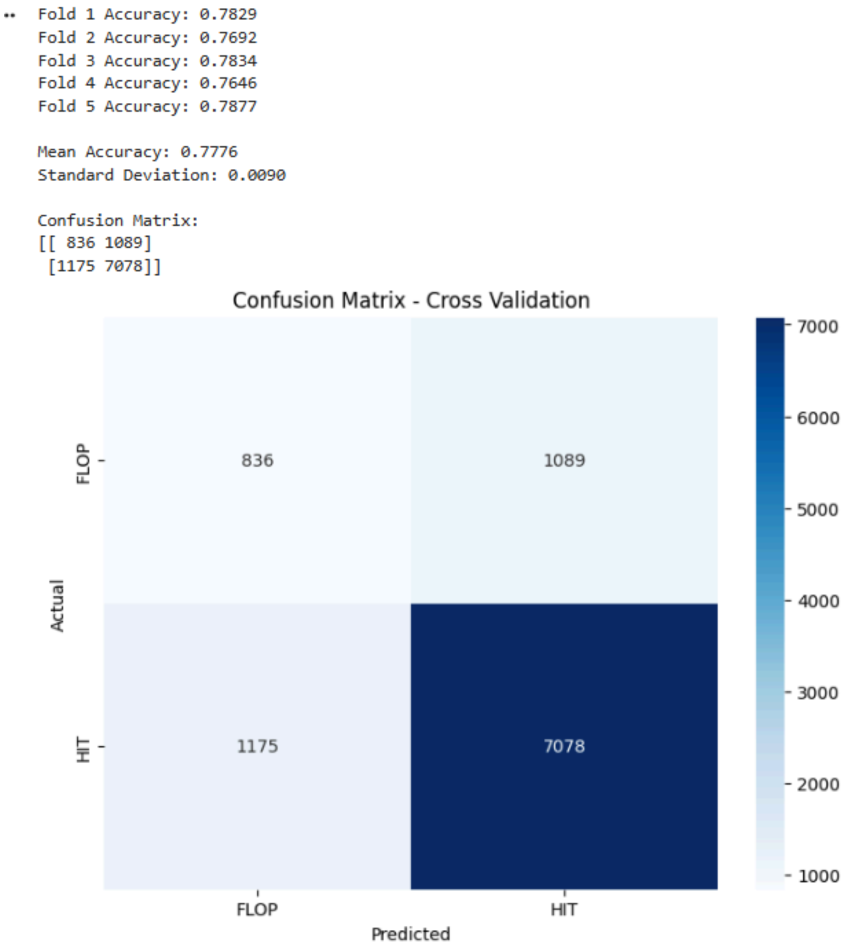
After training the initial decision tree, we observed that the model was able to separate the classes more effectively and provide clearer insight into which attributes were most influential in predicting movie performance.

Although the baseline decision tree performed better than Naïve Bayes, the initial model showed signs of overfitting. Decision trees are prone to growing too deep, memorizing the training data, and losing generalizability. To address this, we applied a series of tuning procedures to optimize the model.

Key hyper parameters we tuned included:

- max-depth: to limit how deep the tree can grow and prevent excessive splitting
- min_samples_split: to ensure that splits only occur when enough data points are present
- min_samples_leaf: to avoid creating branches that rely on very small subsets of data
- criterion = “entropy”: selected to maximize information gain and produce more meaningful splits
- class_weights

After tuning, we obtained an optimized decision tree that demonstrated many things. We got higher accuracy compared to both the baseline tree and Naive Bayes. Reduced overfitting due to controlled depth and minimum sample thresholds.



```
Fold 1 Accuracy: 0.7829
Fold 2 Accuracy: 0.7692
Fold 3 Accuracy: 0.7834
Fold 4 Accuracy: 0.7646
Fold 5 Accuracy: 0.7877

Mean Accuracy: 0.7776
Standard Deviation: 0.0090

Confusion Matrix:
[[ 836 1089]
 [1175 7078]]
```

We ran 5-fold cross validation to ensure accuracy of our model. From the confusion matrix, we can see that the model correctly identifies a large majority of successful movies, with 7,078 true positives. This demonstrates that the classifier is highly sensitive to the characteristics of films that are “hits”. The model correctly identifies 836 flops that are true flops, but it misclassifies 1,089 flops as hits. This might be because the movie's features are overlapping with “hits” making them more difficult to distinguish. These outcomes are to be expected since we had an imbalance dataset.

Across the five folds, model accuracy remained highly consistent, ranging from 0.7877, with a mean accuracy of 0.7776 and a very small standard deviation of 0.0090. This indicates that the model is stable and performs reliably across different subsets of the data. The relatively high mean accuracy suggests that our decision tree is effective at capturing the underlying patterns associated with move performance outcomes.

4.4 Decision Tree Limitations

Although the decision tree classifier outperformed the Naïve Bayes model and provided meaningful insights into the factors associated with movie success, several limitations should be acknowledged. Decision trees are highly sensitive to the structure of the training data. Small changes can dramatically change the tree structure. This instability can reduce the consistency and reliability of predictions, especially when working with datasets that contain noise, missing values, or imbalanced classes.

For our dataset, there was an imbalanced issue between “hit” and “flop” movies in the dataset. This was reflected in the confusion matrix, where flops were more frequently classified as hits. This resulted in the model struggling to capture the characteristics of the flop films reducing effectiveness. When one class is significantly more prevalent than the other, the model becomes biased toward predicting the majority class because doing so minimizes overall error during training. In our case, if the dataset contained far more flop movies than hits (or vice versa), the tree may have learned splitting rules that favored the dominant class, resulting in poor sensitivity to the minority class (*GeeksforGeeks, “How Do Decision Trees Work with Unbalanced Datasets?”*). This imbalance can distort the structure of the tree, reduce the model’s ability to identify subtle patterns associated with rare outcomes, and ultimately lead to misleading performance metrics, such as high accuracy but low recall for the minority class. Even with techniques such as SMOTE or adjusted class weights, decision trees remain susceptible to skewed learning when the underlying distribution of outcomes is uneven (*Matharaarachchi et al.*). This limitation highlights the need for careful preprocessing and evaluation to ensure that predictions remain meaningful and that the model performs effectively across both hit and flop categories.

Lastly, decision trees do not naturally handle high-cardinality categorical variables, such as detailed genre classifications or cast attributes, without extensive preprocessing (*Macro*). This can limit their ability to incorporate rich contextual data unless additional techniques or encoding strategies are applied.

Despite these limitations, decision trees remain valuable due to their interpretability and flexibility, and they provide a strong foundation for more advanced ensemble methods such as Random Forests or Gradient Boosting, which can address many of these challenges.

5. Benchmark and preprocessing

5.1 Benchmarks

(i) Benchmark accuracy without preprocessing

We first set a simple benchmark to understand what level of performance the model can achieve without any pretreatment or learning. When we check our target variable, *hit_or_flop*, we find that each category is not even. Most of the movies in our data set are marked as “hit”, while the proportion of “flop” movies is much smaller.

After counting the viewing data of various films, we found that a total of 8,253 films were hit and 1925 films flop. This means that hit films account for slightly more than 80% of all films. Due to this imbalance, the accuracy rate of a model that predicts "hit" for each film alone has reached about 81.1%. Although the model itself does not have real predictive power, it provides us with a reference benchmark. Any meaningful model must be better than this majority-class-based benchmark.

The class distribution is as follows:

- hit: 8,253 movies
- flop: 1,925 movies

(ii) Class proportions

After creating the binary target variable, *hit_or_flop*, we analyzed the distribution of the two types of data. In the cleaned data set, there are a total of 10,178 movies. Among them, 8,253 films were marked as “hit” and 1,925 films were marked as “flop”. This means that popular movies account for about 81.1% of all movies, while unpopular movies account for only 18.9%. This imbalance is very important for interpreting our research results. Because most movies are popular, a model can get a relatively high accuracy rate by predicting the "hot" of most samples. Based on this, we use the category ratio to set the benchmark, and remember that the actual performance of the model may be overestimated from the overall accuracy alone, especially in the minority category of failed films.

5.2 Pre-processing Steps and Change in Model Accuracy

After determining our target variables, we proceed to prepare the rest of the data sets so that our model can use the data effectively. At this stage, our goal is to sort out the data, transform unstructured fields into available information, and ensure that the model is not confused by text-based or inconsistent input.

One of the problems we solve first is the existence of classified variables stored in text form. For example, columns such as “original language”, “country” and “director” cannot be directly understood by the model, so we have numerically coded them. For these three fields, we use tag coding to keep their structure simple and consistent. For the “Type” column, since a movie can belong to multiple genres, we convert it into a set of virtual variables. In this way, each type can be clearly represented, so that the model can understand the type pattern in a more systematic way.

We have also added some additional features that we think will help the model run. From the list of actors, we extracted the director's name and counted the number of crew members, thus creating a “actor_count” variable. These new features help to convert long text strings into more meaningful numerical information.

After completing the classification coding and feature engineering, we combine all numerical features, such as scores, budgets and the number of actors, with coding variables and type virtual variables to form our final feature matrix. After confirming that there are no missing values, we divide the data into training sets and test sets.

Then, we trained a decision tree classifier with preprocessed data sets. The accuracy rate of this model has reached 82.4%, which is higher than the benchmark accuracy rate of 81.1% in most categories. Although this improvement is not large, it shows that the structured processing of data and the creation of clearer features make the performance of the model better than simply making a “hit” prediction of each film.

5.3 Iterations with Different Pre-processing Steps and Model Comparison

After completing the basic preprocessing, we further tried different feature selection and conversion methods to observe whether the model performance could be improved. Each experiment maintains the same decision tree setting, so that we can focus on understanding the impact of the data itself, not the algorithm itself.

First of all, we tested the effect of eliminating individual features. When we deleted the variable "number of actors", the accuracy of the model decreased slightly to about 81.97%, indicating that the variable provides information of certain value. In contrast, deleting the variable of "director coding" increases the accuracy rate to 83.10%, indicating that director information may introduce interference factors rather than help prediction.

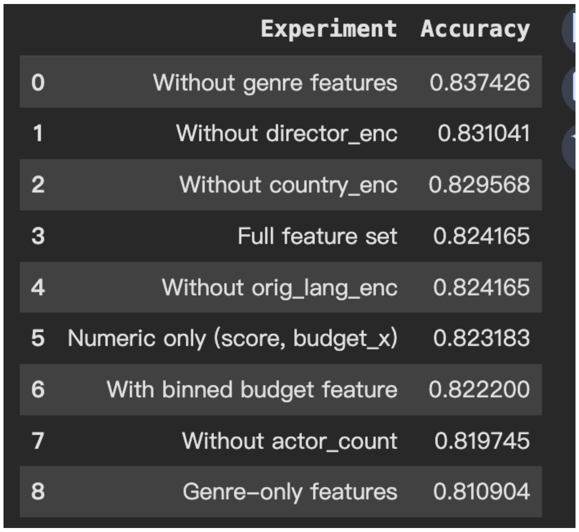
Next, we analyzed the type information. After eliminating all types of virtual variables, the accuracy rate of the model reaches about 83.74%, which is even higher than the complete feature set. This shows that the genre may not be a powerful indicator for judging whether a movie is successful or failing, and sometimes it may even interfere with the judgment of the model.

We also evaluated the effect of removing other classification features. The deletion of the feature of "national code" makes the accuracy rate reach 82.96%, slightly higher than the accuracy rate of the complete model. The deletion of the "original language coding" did not change the accuracy rate, which remained at 82.42%, indicating that the original language may not have a significant impact on the predicted results.

In addition to feature removal, we also tested two simplified feature sets. The accuracy of the model using only numerical variables (score and budget) has reached 82.32%, which is similar to the accuracy of the complete model. This shows that the basic numerical information itself contains strong predictive signals. The accuracy rate of the model using type information alone is relatively low, at 81.09%, which confirms that the type information alone is weaker than the numerical and coding characteristics.

Finally, we divide the budget variables into three categories: low, medium and high. The characteristic accuracy rate after grouping is 82.22%, which is slightly lower than the continuous budget variable. This shows that the original digital budget information is more useful than classifying it.

In general, these experiments show that certain features (especially directors and types) may interfere rather than improve model performance. On the other hand, simple numerical variables such as scores and budgets have the strongest predictive power. This repeated comparison helps us better understand which features should be retained and which features do not contribute significantly to the decision tree model.



	Experiment	Accuracy
0	Without genre features	0.837426
1	Without director_enc	0.831041
2	Without country_enc	0.829568
3	Full feature set	0.824165
4	Without orig_lang_enc	0.824165
5	Numeric only (score, budget_x)	0.823183
6	With binned budget feature	0.822200
7	Without actor_count	0.819745
8	Genre–only features	0.810904

6. Learnings, Takeaways, and Analysis

6.1 Learnings from data-mining analysis

In this project there were several insights that we gained from data mining and how data analytics can be able to help us understand what leads to movie success. We realized that the quality of the data matters a lot. Having missing values and data that is incomplete can lead to inaccurate results. Additionally, we learned how categorical variables need to be encoded in order to be able to use them efficiently in machine learning. Making our label 'hit or flop' was important for us to transform financial data into a useful classification target. Lastly, we realized the importance of data visualizations in order to be able to find patterns and be able to see our data visually with the eye. Looking at our confusion matrix and decision-tree helped us to be able to see which variables were the predictors.

We also learned that our initial model, the Naive Bayes one, performs best when features are independent. Our data had a strong correlation between budget and revenue, genre and score. Also, many important features were continuous with non-Guassian distributions, making Gaussian Naive Bayes less effective. The model produced low accuracy and poor recall for minority classes, showing clear performance limitations. This taught us that model selection must align with the structure of the data, and that more flexible algorithms are needed when relationships are non-linear.

Switching to a decision tree classifier significantly improved our ability to model complex interactions. Decision trees can handle non-linear relationships, mixed data types, and interaction automatically. Setting the entropy criteria helped capture meaningful splits such as budget thresholds, score cutoffs, and genre differences. After turning the hyperparameters, the model produced more interpretable and accurate outcomes. We learned the value of model interpretability as visualizing the tree revealed how specific attributes influenced predictions.

Completing this data-mining analysis taught us not only how to build predictive models but also how to apply them strategically. We learned the importance of selecting appropriate features, transforming feature types, handling imbalanced data, and interpreting model outputs to generate actionable insights for the film industry. Ultimately, our analysis provides a foundation for using data-driven methods to assist studios, investors, and marketers in evaluating a film's likelihood of success.

6.2 Implications and Actions

We were able to find results and insights that movie studios are able to use. With these results studios can be able to make better informed production and investment decisions. Our findings are able to help know whether a movie becomes a hit or flop.

First, we found that the budget is the strongest predictor in deciding whether a movie is a hit. Through our decision tree, we found that the first split was budget. Movies with higher budgets were seen as hits. Therefore movie productions should allocate their budget more towards movies they believe are more promising.

Next, we found audience score to be the second predictor in whether a movie is a hit. The way a movie has its cast, directors, and quality can impact the way an audience scores a movie. This can either negatively or positively impact box-office results and should be taken into consideration for performance.

Moreover, we also found that the country where a movie is produced has an importance in whether a movie is a flop or not. Studios should take into consideration the region where their movies are filmed and consider partnering internationally as well. This should be done in order to produce successful films.

Overall, predictive analytics can help us give greenlighting decisions. As shown our test accuracy is at 82% percent. Therefore this is a strong predictive performance for hits. Movie companies can be able to use data-driven methods in order to be able to financially invest into the best productions as well as be able to prioritize projects that have a higher predicted profitability.

6.3 Other recommendations

Based on our findings we recommend the following steps to be taken:

1. Production studios can create a model in a dashboard that evaluates proposals for new movies. This dashboard can include the estimates of budget, genre, and the expected rating. This can be able to help create a movie that is set to be a hit.
2. If movies are to be predicted as hits or flops before they are released, allocate better marketing and promotional budgeting. Creating marketing campaigns that are stronger can influence the score from the audience and increase profitability.
3. As seen from our analysis, we found that low-budget movies are only a hit when they have strong audience scores or genres that are favorable. Therefore, we recommend that studios should not combine low-budget films with storylines that are weak or inexperienced directors. This may lead to a movie to not hit.

In summary, our recommendations emphasize how predictive analytics can meaningfully support decision-making in the film industry. By integrating data-driven models during the process of making a movie, stakeholders can strategically market resources more, align budget levels with proven audience preferences, and significantly reduce financial uncertainty and improve the likelihood of producing commercially successful films. This can be a powerful tool used for enhancing profitability and guiding more informed, evidence based production strategies and decisions.

References

Associate Vice President, Analytics & Data Strategy. (2024, July 3). Predictive analytics in movies: Using big data to forecast film success. Quantzig. https://www.quantzig.com/blog/predicting-movie-success-data-analytics-film-industry/

C. Burgos, María, et al. "Using Decision Trees to Characterize and Predict Movie Profitability on the US Market ." Proceedings of the International MultiConference of Engineers and Computer Scientists 2015, vol. Vol 1, no. 2078-0966 (Online), 20 Mar. 2015, www.iaeng.org/publication/IMECS2015/IMECS2015_pp274-279.pdf.

Singh, A. (2023). IMDb Movies Dataset [Data set]. Kaggle. https://www.kaggle.com/datasets/ashpalsingh1525/imdb-movies-dataset

Statista. (n.d.). Cinema – Worldwide. Statista. https://www.statista.com/outlook/amo/media/cinema/worldwide

Zhang, Z., [et al.]. (2024). Prediction techniques of movie box office using neural networks an emotional mining. Scientific Reports, 14(1). https://doi.org/10.1038/s41598-024-72340-z

GeeksforGeeks. "Naive Bayes Classifiers." GeeksforGeeks, 3 Mar. 2017, www.geeksforgeeks.org/machine-learning/naive-bayes-classifiers/.

GeeksforGeeks. (2025, July 23). How do decision trees work with unbalanced datasets? https://www.geeksforgeeks.org/machine-learning/how-do-decision-trees-work-with-unbalanced-datasets/

Matharaarachchi, Surani, et al. "Enhancing SMOTE for Imbalanced Data with Abnormal Minority Instances." Machine Learning with Applications, vol. 18, Dec. 2024, p. 100597, https://doi.org/10.1016/j.mlwa.2024.100597.

Macro. "Handling Categorical Features in Decision Trees: A Comprehensive Guide to Supervised Learning With…." Medium, 27 Nov. 2024, medium.com/@jangdaehan1/handling-categorical-features-in-decision-trees-a-comprehensive-guide-to-supervised-learning-with-c1bfa1783ea7.